

Mobiilipelin toteuttaminen ja sillä ansaitseminen

Janne Hiltunen

Opinnäytetyö
Toukokuu 2015

Mediatekniikan koulutusohjelma
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) Hiltunen, Janne	Julkaisun laji Opinnäytetyö	Päivämäärä 18.5.2015
	Sivumäärä 58	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Mobiilipelin toteuttaminen ja sillä ansaitseminen		
Koulutusohjelma Mediatekniikan koulutusohjelma		
Työn ohjaaja(t) Manninen, Pasi		
Toimeksiantaja(t) tmi Janne Hiltunen		
<p>Tiivistelmä</p> <p>Opinnäytetyö ja siihen liittyvä projekti toteutettiin opinnäytetyön tekijän omalle toiminimelle. Opinnäytetyöllä tavoiteltiin kokemusta pelienteon prosessista, sekä kannustusta kokonaisen pelin toteuttamiseen.</p> <p>Opinnäytetyössä tutkittiin eri pelimoottoreita, joilla voidaan toteuttaa mobiilipelejä. Vertailuun otettiin neljä toisistaan hieman erilaista pelimoottoria: Unreal Engine 4, Unity 3D, Clickteam Fusion 2.5 ja Game Maker: Studio. Näille pelimoottoreille tehtiin vertailua eri osa-alueista, kuten graafisista ominaisuuksista, käytettävyydestä sekä ohjelmointikielistä. Vertailun tuloksena valittiin opinnäytetyön projektille pelimoottoriksi Unity 3D, suurilta osin sen helppokäyttöisyyden sekä kattavan kehittäjäyhteisön ansiosta.</p> <p>Opinnäytetyössä esiteltiin pelintekoa prosessina, sekä käytiin läpi opinnäytetyötä varten aloitetun pelin suunnitelmat. Pelinteon prosessia tutkiessa käytiin läpi mahdolliset huomioon otettavat ongelmakohdat. Opinnäytetyötä varten aloitettu peli sai työnimen Medieval Battlefield. Tämän pelin kehitystä jatketaan kesällä ja syksyllä 2015. Peli on vuoropohjainen strategiapeli mobiilialustoille.</p> <p>Opinnäytetyössä tutkittiin pelien eri ansaintamenetelmiä. Ansaintamenetelmät jaettiin kolmeen eri kategoriaan: kertamaksuiset, kausimaksuiset ja sisältömakuiset. Tutkimuksessa selvitettiin muun muassa millä ansaitsee parhaiten ja mikä on käyttäjäystävällisin tapa. Medieval Battlefieldin tapauksessa ansaintamenetelmänä päädyttiin käyttämään sisältömaksuisuutta.</p> <p>Opinnäytetyössä päästiin haluttuun tavoitteeseen, ja pelin kehitystä tullaan jatkamaan opinnäytetyön jälkeen.</p>		
Avainsanat (asiasanat)		
Mobiilipeli, peliohjelmointi, pelisuunnittelu, ansaintamenetelmät		
Muut tiedot		



Description

Author(s) Hiltunen, Janne	Type of publication Bachelor's thesis	Date 18.5.2015
		Language of publication: Finnish
	Number of pages 58	Permission for web publication: x
Title of publication Creating a mobile game to earn money		
Degree programme Media Engineering		
Tutor(s) Manninen, Pasi		
Assigned by Hiltunen, Janne		
<p>Abstract</p> <p>The thesis and the game project associated with it were assigned by the author's company name. The purpose of the thesis was to build up experience of the process of game making as well as to work as an incentive to make a full game.</p> <p>Different game engines able to build mobile games were investigated. A total of four slightly differing game engines were chosen for comparison: Unreal Engine 4, Unity 3D, Clickteam Fusion 2.5 and Game Maker: Studio. These engines were compared using different categories, e.g. their graphical capabilities, usability as well as programming languages. From these four game engines Unity 3D was chosen as the game engine for the project, mostly due to its usability and vast developer community.</p> <p>The process of game making is discussed in the thesis, and the plans for the project itself were also briefly studied. The possible problems to consider were viewed as well in the game making process. The game that was started for the thesis was given the work title "Medieval Battlefield". This development of this game will be continued till fall 2015. The game is turn based a strategy game for mobile platforms.</p> <p>Different kind of earning methods were looked at. These earning methods were divided into three categories: one-time payment, season based billing and content based billing. These categories were compared to see, for example, which method helped to earn most money and which one was the most user friendly. In the case of Medieval Battlefield, the content based billing was chosen.</p> <p>The thesis achieved the required objectives and the development of the game will be continued after the thesis.</p>		
Keywords/tags (subjects) Mobile game, game programming, game planning, earning methods		
Miscellaneous		

Sisältö

1 Työn taustat	4
1.1 Tausta ja toimeksiantaja	4
1.2 Tehtävä ja tavoitteet	4
2 Taustatutkimus	6
2.1 Pelimoottorit	6
2.1.1 Unity 3D	6
2.1.2 Unreal engine 4	10
2.1.3 Clickteam fusion 2.5	13
2.1.4 Game Maker: Studio	15
2.2 Vertailun tulokset	17
3 Peli-idea	20
3.1 Pelin tavoite	20
3.2 Pelin mekaniikat	20
3.2.1 Taistelu	21
3.2.2 Armeijat	23
3.2.3 Kasarmit ja väsyminen	23
3.2.4 Kokemustasot	23
3.2.5 Taitotaso	24
3.3 Joukkotyypit	24
3.3.1 Yleistä	24
3.3.2 Footman	25
3.3.3 Archer	26
3.3.4 Knight	26
4 Peli-ideasta peliksi	27
4.1 Pelinteko prosessina	27
4.1.1 Suunnittelu	27
4.1.2 Toteuttaminen	29
4.1.3 Julkaisu	31

4.2 Testausmenetelmät mobiilipelissä.....	31
4.3 Grafiikan toteuttaminen peliin.....	35
4.3.1 Graafisen käyttöliittymän ja käyttöliittymäkokemuksen toteuttaminen ..	35
4.3.2 3D mallien mallintaminen	39
5 Ansaitseminen pelillä	44
5.1 Pelien eri ansaintamenetelmät	44
5.1.1 Kertamaksuiset pelit.....	44
5.1.2 Kausimaksuiset pelit.....	44
5.1.3 Sisältömaksuiset pelit.....	45
5.2 Mikä on käyttäjäystävällisin tapa ansaita pelillä	47
5.3 Millä tavalla ansaitsee parhaiten?	47
5.4 Oman pelin ansaintaperiaate	54
5.4.1 Freemium-menetelmä	54
5.4.2 Mainokset.....	55
6 Pohdinta	56
Lähteet	58

Kuviot

Kuvio 1. Kuvakaappaus Unityn työtilasta	7
Kuvio 2. Esimerkki Unity 3D:n graafisista kyvyistä	10
Kuvio 3. Unreal Engine 4:n käyttöliittymä.....	12
Kuvio 4. Esimerkki Unreal Engine 4:n graafisista kyvyistä	13
Kuvio 5. Clickteam Fusion 2.5 event -editor	14
Kuvio 6. Game Maker: Studion käyttöliittymä.....	16
Kuvio 7. Esimerkki taistelun ensimmäisestä kierroksesta.....	22
Kuvio 8. Footman varustettuna miekalla ja kilvellä	25
Kuvio 9. Esimerkkejä yksinkertaisesta graafisesta tyylistä.....	29
Kuvio 10. Unity profiler	33
Kuvio 11. Suunniteltu hierarkia eri näkymien välillä.....	37

Kuvio 12. Armeijan suunnittelunäkymän luonnos	38
Kuvio 13. Armeijan suunnitteluikkunan lopullinen grafiikka	39
Kuvio 14. Rautalankamallinnos pelin footman-hahmosta	41
Kuvio 15. UVW mappaus 3ds maxissa	42
Kuvio 16. Skinnattu footman.....	43
Kuvio 17. Guild Wars 2 sisältöostokyselyn tulokset	49
Kuvio 18. Clash of Clans pelaajien käyttämät rahasummat.....	51
Kuvio 19. Eri ansaintamuotojen pelien tuomat tulot aloitusmaksuista tuhatta henkilöä kohden	52
Kuvio 20. Eri ansaintamuotojen tuomat tulot vuoden ajalta tuhatta henkilöä kohden (huomioi logaritminen asteikko)	53
Kuvio 21. Pelien vuositulot koko pelaajamäärällä	54

Taulukot

Taulukko 1. Pelimoottorien vertailutulokset (Painoarvo merkitty sulkeisiin)	18
---	----

1 Työn taustat

1.1 Tausta ja toimeksiantaja

Medieval Battlefielдин kehitys aloitettiin kehittäjän eli tämän opinnäytetyön tekijän halusta ja intohimosta kehittää peli ja saattaa se julkaisuun asti. Kehittäjällä on aikaisemmin ollut useita peliprojekteja, mutta nämä pelit on jouduttu unohtamaan lähinnä ajan puutteen vuoksi. Tekemällä opinnäytetyön pelinteen prosessista kehittäjä haki kannustusta saada peli valmiiksi julkaistavaksi, ja tätä kautta tärkeää kokemusta sekä referenssi työnhakua varten.

Peliprojektin tilaajana toimi tämän opinnäytetyön kirjoittajan oma toiminimi. Toiminimi on perustettu vuoden 2013 joulukuussa perustajan tarpeesta toimia virallisesti tulevien asiakkaiden kanssa.

1.2 Tehtävä ja tavoitteet

Medieval battlefield (työnimi) on mobiilialustoille toteutettava peli. Opinnäytetyön tavoite oli luoda tietotaitopohja pelien kehittämiseen projektin toteuttajalle. Tavoite oli luoda toimiva pohja pelille opinnäytetyötä tehdessä, josta voidaan sittemmin kehittää kesä-syky 2015 aikana lopullinen julkaistava versio pelistä. Projektista toteutettiin tätä opinnäytetyötä varten versio, jonka pohjalta voitiin osittain kirjoittaa luku ”4 Peli-ideasta peliksi”.

Tutkittaessa pelin erilaisia ansaintamenetelmiä joudutaan kuitenkin tukeutumaan enemmissä määrin internetistä löytyvään tutkimustietoon, koska peliprojektia ei ole järkevää julkaista keskeneräisenä, ja vertailla sillä saatuja tuloksia. Näitä ansaintamenetelmiä tutkiessa tavoiteltiin tietoa siitä, mikä olisi opinnäytetyön projektin kannalta paras tapa ansaita.

Peli toteutetaan Unity 3D:llä, koska se on todettu nopeaksi ja helpoksi kehitysympäristöksi, lisäksi sen ympärillä on suuri kehittäjäyhteisö, jolta voi kysyä tarvittaessa apua. Muitakin vastaavia kehitysympäristöjä tutkitaan opinnäytetyössä, mutta vain vertailua tehtäessä.

2 Taustatutkimus

2.1 Pelimoottorit

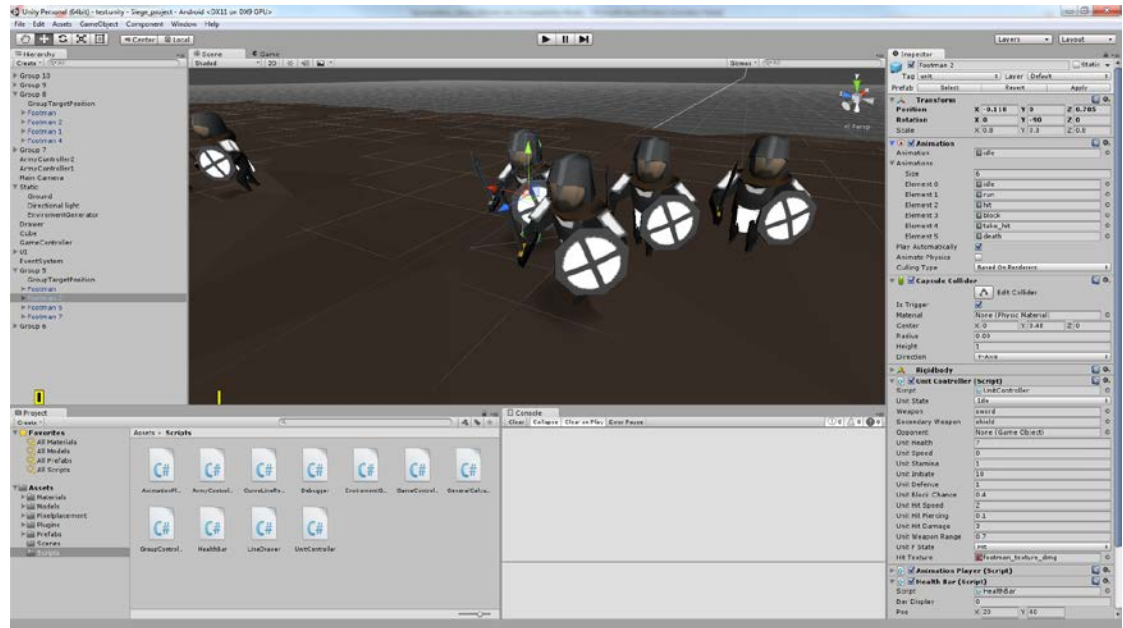
Opinnäytetyössä vertailtiin muutamia tuttuja pelimoottoreita, joista kaikista on mahdollisuus myös tuoda sovellus mobiilialustalle. Vertailussa otettiin huomioon moottoreiden sekä niiden kehitysohjelmien käytössä olevat kielet, graafiset ominaisuudet sekä helppokäyttöisyys. Koska aika ei riittänyt perusteelliseen tutkimukseen ohjelmien sekä moottorien välillä, on vertailussa pitkälti myös nojaututtu muiden käyttäjien kokemuksiin ohjelmista.

2.1.1 Unity 3D

Unity on Unity Technologiesin kehittämä pelimoottori, joka tukee useita alustoja, muun muassa PC, Android, iOS, Playstation (3 ja 4), xbox (360 ja one). Unityn ensimmäinen versio julkaistiin vuoden 2005 kesäkuussa. Unityn ensimmäinen versio tuki tuontia vain Mac OS:lle, ja Unity saikin vasta versiossa 2.0 Windows:n sekä iOS:n tuen. Lokakuussa 2010 julkaistiin Unityn versio 3.0, jossa merkittävimpiä päivityksiä oli erikseen ostettava Android tuki. Unity 4.0 julkaistiin vuoden 2012 syksyllä, ja tätä versiota voisi kutsua Unityn suosion aluksi. Viimeisin 5.0-versio julkaistiin 2015 maaliskuussa, jolla myös tämä projekti on toteutettu. Unity 5:n myötä Android, iOS sekä Windows mobile tuki tuli ilmaiseksi Unitylle. Aiemmin julkaisulisenssi näille oli maksanut 500 euroa. (Unity (game engine) 2015)

Unity 3D on vertailluista moottoreista käytetyin indie-kehittäjien keskuudessa. Tästä syystä sille on myös helpointa löytää teknistä tukea, mikäli törmää ongelmiin. Suuren suosionsa ansiosta Unity saa koko ajan lisää käyttäjiä. Tämä taas johtaa hyvään tukeen kehittäjiltä ja karsimattomiin päivityksiin. Unityn nykyversion käyttäminen on indie-kehittäjälle ilmaista, mutta muuttuu maksulliseksi, kun ohjelmalla tehdyn pelin tuotot nousevat yli 100 000 \$ vuodessa.

Unityn käyttöliittymä on selkeä ja looginen. Sitä oppii käyttämään kuka tahansa vastaavia ohjelmia käyttänyt esimerkiksi katsomalla yhden lukuisista opasvideosta. Käyttöliittymä koostuu viidestä osasta, jos ei lasketa päälle tuotavia lisäosien ikkunoita tai MonoDevelop-koodikehitysympäristöä. Unityn käyttöliittymän voi järjestää haluamaansa järjestykseen. Kuviossa 1 nähdään vasemmalla hierarkiapaneeli, keskellä scene/peli, oikealla inspector, alhaalla vasemmalla kansiorakenne ja keskellä alhaalla konsoli.



Kuvio 1. Kuvakaappaus Unityn työtilasta

Hierarkiapaneelissa ovat kaikki sen hetkisen scenen objektit. Objekteja voidaan asettaa hierarkiassa toistensa vanhemmiksi tai lapsiksi, jolloin voidaan objektien koodissa viitata sen objektin vanhempaan tai lapseen. Mikäli objektin vanhempaa liikutetaan pelimaailmassa, liikkuu myös lapsi mukana saman etäisyyden, lasta liikuttaessa ei vanhempi kuitenkaan liiku.

Scene-näkymässä pelaaja voi liikuttaa kameraa vapaasti 3-ulotteisessa tilassa. Tässä tilassa näkyvät kaikki sen hetkisen scenen objektit niille määrätyillä x-, y- ja z-koordinaateillaan. Käyttäjä voi halutessaan valita objektin suoraan scenestä. Scene-näkymä tuo käyttäjälle myös lisätietoa gizmojen muodossa. Nämä gizmot voivat

näyttää mm. eri objektien kiihtyvyyttä, orientaatiota tai valon lähdettä. Scene-näkymän toisena välilehtenä on peli-näkymä, jossa peliä voidaan koeajaa. Scene- ja peli-näkymän välillä voidaan vaihdella vapaasti, kun peliä koeajetaan.

Inspector-näkymässä nähdään sen hetkisen valitun objektin komponentit. Jokaisella Unity objektilla on aina transform-komponentti, eikä sitä voida poistaa. Kaikki muut komponentit ovat käyttäjän valittavissa. Objekteille voidaan asettaa eri komponentteja, kuten scriptejä, fysiikka-komponentteja, animaattoreita yms. Jos script objektissa olevalle scriptille asetetaan julkinen muuttuja, voidaan inspector näkymässä suoraan asettaa sen objektin julkinen arvo.

Kansio-näkymä näyttää projektin kansion kansiorakenteen, ja kansio-näkymän kautta voidaan myös luoda scriptejä, materiaaleja ja shadereita. Kun kansio-näkymästä valitaan tiedosto, sen tiedot tulevat näkyville oikean laidan inspector-näkymään.

Alareunan konsoliin tulee Unityn ilmoittamat virheet koodissa. Käyttäjä voi myös itse debug komennolla tuoda tähän tarpeellista tietoa koodistaan, tai kokeilla esimerkiksi, suorittaako ohjelma ikinä haluttua funktiota.

Unity engine tukee kolmea eri ohjelmointikieltä: C#, UnityScript ja Boo. Näistä viimeinen on poistumassa käytöstä Unitystä, mutta on yhä tuettu. Tämä ohjelmointikielituki on kattavin vertailuun otetuista moottoreista. Itse ohjelmointi ei tapahdu Unityssä, vaan Unityn mukana tulee MonoDevelop-ohjelmointieditori, jossa on syntaksin täydennys valmiiksi asennettuna kaikille Unityn tukemille kielille.

UnityScript on JavaScriptin kaltainen dynaamisesti tyyplitetty ohjelmointikieli, ja on siksi helposti lähestyttävä käyttäjille, jotka ovat kokeneita verkko-ohjelmistokehittäjiä. Boo on syntaksiltaan Pythonia muistuttava kieli, eli siinä ei käytetä juuri laisinkaan sulkeita tai välimerkkejä. Seuraavan sivun esimerkeissä on toteutettu sama ohjelma kaikilla Unityn tukemilla kielillä. Unityn tarjoamista ohjelmointikielistä Unityscript ja C# ovat käytetyimpiä, ja niille löytyy internetistä tarvittaessa runsaasti tukea. Opinnäytetyön projektissa oli käytössä C# lähinnä

toteuttajan omasta mieltymyksestä ”rautaläheisempään” ohjelmointikieleen. Kattava ohjelmistokielituki on hyvä valttikortti Unitylle tässä vertailussa, ja tästä syystä se sijoittuu parhaiten tässä osavertailussa.

C#:

```
using UnityEngine;
using System.Collections;

public class ExampleSyntax : MonoBehaviour {
    int myInt = 5;

    int MyFunction (int number) {
        int ret = myInt * number;
        return ret;
    }
}
```

Javascript (Unityscript):

```
#pragma strict

var myInt : int = 5;

function MyFunction (number : int) : int {
    var ret = myInt * number;
    return ret;
}
```

Boo:

```
import UnityEngine
import System.Collections

public class ExampleSyntax(MonoBehaviour):

    private myInt = 5

    private def MyFunction(number as int) as int:
        ret as int = (myInt * number)
        return ret
```

Unityn grafiikkamoottori on nykystandardeilla ajantasainen. Se tukee DirectX 11:ta joka takaa sen grafiikkamoottorille kykyä tehdä monipuolisia shadereita sekä kuvaefektejä. Unityllä voidaan tehdä myös täysin 2-ulotteisia pelejä, joissa voidaan myös hyödyntää kaikkia DirectX 11:a tuomia ominaisuuksia. Käyttäjä pystyy itse tekemään Unityyn uusia shadereita, tämä antaa käyttäjälle haluttua vapautta. Internetistä löytyy myös lähes joka käyttöön mieleinen shader, mikäli käyttäjä ei itse osaa tai halua tehdä shaderia. Unityn graafiset ominaisuudet ovat vertailuun otetuista ohjelmista toiseksi parhaat. Kuviossa 2 nähdään esimerkki Unity 3D:n graafisista kyvyistä.



Kuvio 2. Esimerkki Unity 3D:n graafisista kyvyistä (p.a.m.e.l.a. 2015)

2.1.2 Unreal engine 4

Unreal engine 4 on ehkä vertailluista pelienkehitysohjelmista vakavimmin otettava. Sen historia ulottuu vuoteen 1998, kun moottorin ensimmäinen versio julkaistiin. Verrattuna muihin aikansa pelimoottoreihin, se tarjosi paljon ominaisuuksia pakattuna yhteen moottoriin. Vuonna 2002 Unreal Enginestä julkaistiin versio 2. Uusi versio toi lähinnä parannuksia moottorinrenderöinti ominaisuuksiin ja tukea uusille alustoille, kuten GameCubelle sekä Xboxille. Unreal Enginen kolmas versio julkaistiin

vuonna 2004 reilun 18 kuukauden kehityksen jälkeen. Tämäkin päivitys alkuun keskittyi lähinnä piirtomoottorin päivityksiin. Alkuun Unreal Engine 3 tuki vain Windows-käyttöjärjestelmää, Playstation 3:a sekä xbox 360:tä. Vuonna 2010 se sai tuen Androidille sekä iOS:lle, joten sillä pystyttiin kehittämään pelejä myös mobiilialustoille. Unreal Enginen uusin versio, 4, julkaistiin vuonna 2012, tosin sitä oli kuitenkin kehitetty salassa jo vuodesta 2003. (Unreal Engine 2015)

Uusimmassa versiossa on otettu huomioon indie-kehittäjät sen lisenssimaksuissa, Unreal Engine 4:ää on ilmaista käyttää kaupallisessa käytössä niin kauan, kuin sillä kehitetyn tuotteen neljännesvuositulot ovat alle 3000 \$. Tämän jälkeen Unreal Enginen kehittäjät haluavat 5 % tuotosta. Unreal Engine 4 tukee lähes kaikkia nykyaikaisia alustoja, mobiilialustoista ei kuitenkaan Windows ole tuettu.

Huomioitavaa Unreal Enginen arvioinnissa on, että arviointia varten Unreal Enginen kehitystyökalujen asentaminen ei onnistunut, joten arvostelut perustuvat täysin internetistä löytyneisiin käyttäjäkokemuksiin.

Unreal Engine 4:n käyttöliittymä on hyvin samankaltainen kuin Unity 3D:llä. Se koostuu käytännössä samoista komponenteista kuin Unity. Tästä syystä Unity-kehittäjän on helppo siirtyä Unreal Engineen tai päinvastoin. Kuten jo aiemmin on Unityn käyttöliittymä todettu helposti opittavaksi, on loogisesti myös Unreal Enginen käyttöliittymä nopea oppia ja selkeä käyttää. Kuviossa 3 on kuvakaappaus Unreal Engine 4:n käyttöliittymästä.



Kuvio 3. Unreal Engine 4:n käyttöliittymä (Tpelgam42 2014)

Unreal Engine 4:n ainoa tukema kieli on C++, eikä käyttäjällä ei ole muita vaihtoehtoja. Aiemmin Unreal Enginet käyttivät omaa skriptikieltään, UnrealScriptiä. Uusimmassa versiossa se kuitenkin vaihdettiin kokonaan C++:aan. Vaikkakaan Unreal Engine 4:llä ei ole tukea useille kielille, on C++ silti hyvä valinta. Se on helppo oppia aloittelijoidenkin ja se on riittävän ”rautaläheistä” ohjelmointikieltä tarjoten myös kokeneemmille kehittäjille tarpeeksi vapautta.

Graafisesti Unreal Engine 4 on ehdottomasti vertailussa olleista moottoreista kehittynein. Se tukee viimeisintä DirectX versiota ja sen myötä kehittyneitä shadereita, kuten tesselaatiota, dynaamista valaistusta sekä monia muita graafisesti kauniita ominaisuuksia. Näistä syistä Unreal Engine nousee selkeästi kärkeen sen graafisissa ominaisuuksissa. Kuviossa 4 on kuvakaappaus vuoden 2015 alussa julkaistusta teknologiademosta.



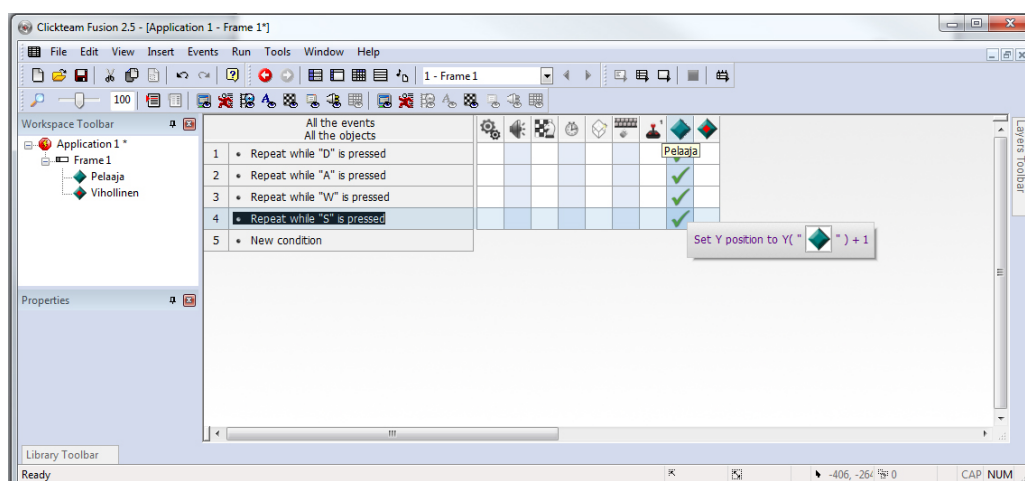
Kuvio 4. Esimerkki Unreal Engine 4:n graafisista kyvyistä (Dereau Benoit 2015)

2.1.3 Clickteam fusion 2.5

Clickteam fusion 2.5 on joukon ”musta lammas”. Se ei ole niinkään perinteinen ohjelmointiin pohjautuva pelimoottori, vaan pelien ”ohjelmointi” tapahtuu ns. eventien kautta. Clickteam fusion 2.5 on saanut alkunsa vuonna 1994 Clickteam nimisen yrityksen julkaisemasta Klik & Play pelienkehitysohjelmasta. Klik & Play tarjosi aikanaan helpon tavan ohjelmoinnista kiinnostuneille opetella ohjelmoinnin logiikkaa sen helpolla event-pohjaisella ohjelmointimetodilla. Myöhemmin, vuonna 1996, Clickteam julkaisi uuden pelientekosovelluksen: The Games Factoryn. The Games Factory toi mukanaan uusia ominaisuuksia sekä hieman päivitetyn käyttöliittymän. The Games Factory oli 2000 luvun alussa erittäin suosittu suomalaisten nuorten pelienkehittäjien joukossa, ja tuohon aikaan olikin useita yhden miehen pelistudioita internetissä. Vuonna 1998 julkaistiin taas uusi versio Clickteamin tuotesarjaan: Multimedia Fusion. Se oli Clickteam Fusion 2.5:n edeltäjä ja tarjosi mm. ohjelmoitavat laajennukset. Clickteam Fusion 2.5 julkaistiin lopulta vuonna 2006, ja se tarjosi edeltäjiensä lisäksi mm. grafiikkatasot, Java-mobiilipeli exportin sekä uuden kuva- ja animaatioeditorin. (Clickteam 2015)

Vaikka Clickteam Fusion 2.5 on jo osittain melko ikääntynyttä ohjelmistoa, se on yhä melko suosittu pelienkehitystyökalu johtuen sen erilaisesta, helposta lähestymistavasta ohjelmoida pelejä. Clickteam Fusion 2.5 saa yhä hyvää tukea sen käyttäjiltä sekä kehittäjiltä C++:lla ohjelmoitavien laajennusten ansiosta. Clickteam fusion 2.5 integroi pelimoottoriin aiemmin erillisenä laajennuksena olleen Box2d-fysiikkamoottorin. Clickteam Fusionista voidaan tuoda peli suoraan html5-muotoon, ja siihen on mahdollista ostaa laajennuksena iOS-, Android- ja Flash-vientimoduulit.

Kuten aiemmin jo on mainittu, Clickteam Fusion 2.5 eroaa merkittävästi muista vertailussa olevista ohjelmista sen helposti lähestyttävän ohjelmointitavan ansiosta. Clickteam Fusionissa ei varsinaisesti ohjelmoida peliä rivi riviltä, vaan ohjelmointi perustuu taulokkolaskennan kaltaiseen taulukkoon, jossa riveittäin on ”jos tapahtuu”, ja sarakkeeseen lisätään tapahtuma sen objektin kohdalle, jolle halutaan joltain tapahtuvan seurauksena. Yhdellä tapahtumalla voi olla useita seurauksia. Nämä tapahtumat voivat olla esimerkiksi objektin siirtämistä tietyllä akselilla, sen ominaisuuksien muutoksia, äänien soittamista tai resoluution vaihtamista. Clickteam Fusion on hyvä tapa opetella ohjelmoimaan, mikäli aiempaa kokemusta ei ole. Sen avulla oppii ymmärtämään ohjelmoinnin perusperiaatteita, ja se antaa hyvän pohjan jollekin kehittyneemmälle ohjelmointikielelle. Kuviossa 5 on esimerkkitoteutus yksinkertaisesta 4-suuntaisesta hahmonliikutusmoottorista.



Kuvio 5. Clickteam Fusion 2.5 event -editor

Clickteam Fusionin käyttöliittymän on helppo omaksua, ja se koostuu pääasiassa kolmesta eri ikkunasta joiden välillä käyttäjä liikkuu. Storyboard editorissa käyttäjä voi luoda useita frameja, jotka toimivat tässä tapauksessa samalla tavalla kuin Unityn Scenet toimivat. Frame editorissa käyttäjä voi muokata yksittäistä framea ja sijoittaa siihen objekteja ja liitännäisiä haluamallaan tavalla. Event editor ikkunassa käyttäjä voi muokata kysyisen framen tapahtumia. Clickteam Fusionin käyttöliittymä on nopeasti opittava ja selkeä kokonaisuus.

Graafisilta ominaisuuksiltaan Clickteam Fusion on tasoltaan samaa luokkaa kuin Game Maker, se kykenee tuottamaan yksinkertaista 3D-grafiikkaan lisäosilla, mutta on kuitenkin pääsääntöisesti suunnattu 2-ulotteisten pelien ja ohjelmien toteuttamiseen. Ohjelman grafiikkaa voidaan luoda sekä animoida Clickteam Fusionin omassa grafiikka-editorissa. Grafiikalle voidaan antaa alpha-taso, jolla voidaan tehdä monipuolisempaa grafiikkaa, sekä eri blend-efektejä kuten lisäävä, vähentävä tai negatiivi. Näiden efektien ja animaatioiden ansiosta Clickteam Fusionilla voidaan osaavissa käsissä luoda näyttäviäkin pelejä.

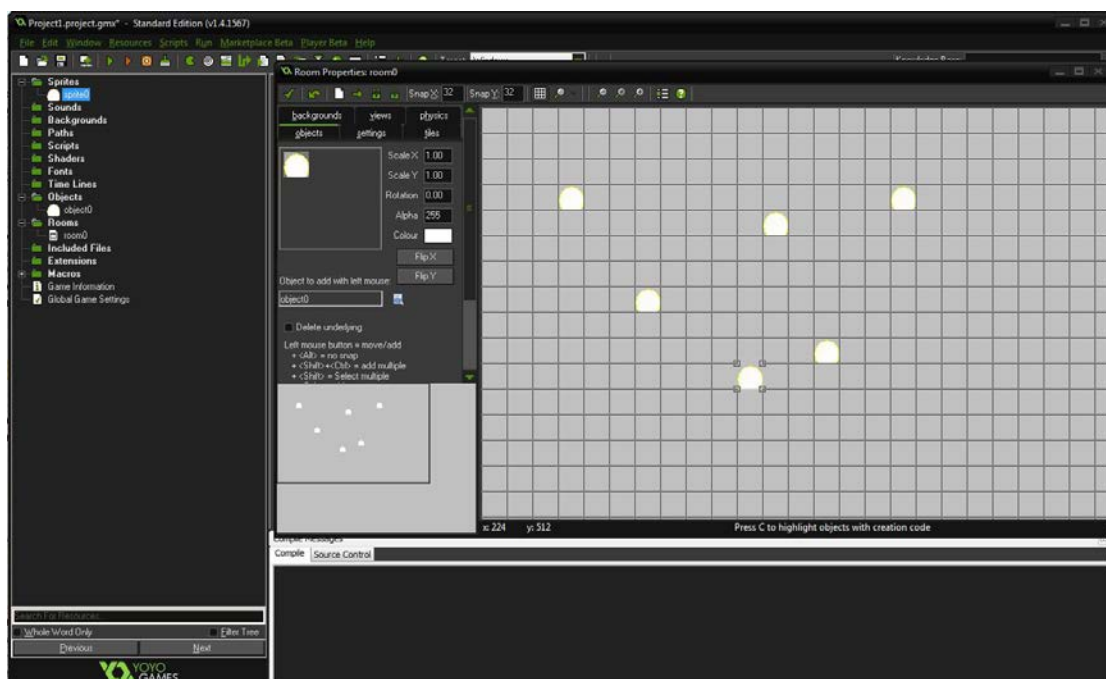
2.1.4 Game Maker: Studio

Kuten Clickteam Fusion, Game Maker: Studio on myös suunniteltu helpoksi lähestymistavaksi pelien tekoon. Sen ohjelmointi perustuu raahaa & pudota logiikkaan. Game Maker: Studio on saanut alkunsa vuonna 1999 Animo nimisestä ohjelmasta, joka oli suunnattu 2D animaatioiden luomiseen. Nimi vaihdettiin myöhemmin GameMakeriksi. Alkuun GameMakerissa ei ollut ollenkaan mahdollisuutta tehdä ajettavia tiedostoja, vaan pelien piti pyöriä GameMakerin omassa toistosovelluksessa. (GameMaker: Studio 2015)

Nykyversiossa Game Maker luo oman ajettavan tiedoston, joka on kaikille sovelluksille sama. Tämä ajettava tiedosto lukee erillisestä data tiedostosta resurssit jotka pyörii sovelluksessa samoin kuin itse Game Makerin sisällä pyörittäessä.

Käytännössä se siis luo kevyemmän version itse kehitysovelluksesta, ilman kehitysominaisuuksia. Game Maker: Studiosta pystytään tuomaan sovelluksia useisiin eri alustoihin, ja onkin yksi kattavimmista vertailuun otetuista sovelluksista. Kuitenkin vain pieni osa tuetuista alustoista on Game Maker: Studion ilmaisversion käytössä.

Game Maker: Studion käyttöliittymä ei ole ehkä helpoimmasta päästä, sillä se on lähes sama kuin se on ollut julkaisusta saakka. Kaikki sen ominaisuudet on pitkälti piilossa useiden valikkojen takana, joten ensikertalaisella menee hieman pidempi aika omaksua Game Maker: Studion käyttöliittymä. Kuviossa 6 nähdään kuvakaappaus Game Maker: Studion käyttöliittymästä.



Kuvio 6. Game Maker: Studion käyttöliittymä

Game Maker: Studiolla ohjelmoiminen on hieman samantapaista kuin Clickteam Fusionilla. Se perustuu kuvaikkeilla kuvattuihin seurauksiin sekä tapahtumiin, joita raahataan oikealta sarakkeesta tapahtumankuuntelijaan. Nämä seuraukset voivat olla esimerkiksi objektin siirtämistä pikselin verran oikeaan, sen tuhoaminen tai vaikka muuttujan muuttaminen. Game Maker: Studiolla voidaan myös ohjelmoida erillisiä laajennuksia, jotka laajentavat ohjelmiston hieman suppeaa

tapahtumakirjastoa. Ohjelma tukee myös valmiiden tapahtumakuuntelijoiden lisäksi omaa GML (Game Maker Language) skriptikieltä, jota voidaan käyttää, kun halutaan hallita paremmin, mitä seurauksia tapahtuu.

Graafisesti Game Maker: Studio pärjää vertailuun otetuista pelinkehitystyökaluista huonoiten. Vaikkakin se kykenee parempaan 3D-grafiikkaan kuin Clickteam Fusion, on sen pääpainotteinen 2-ulotteinen grafiikkatuki suhteellisen heikko.

2.2 Vertailun tulokset

Vertailu pelinkehitysohjelmien sekä pelimoottoreiden välillä tehtiin kolmessa eri kategoriassa: graafiset ominaisuudet, ohjelmointikielet sekä käyttäjäystävällisyys. Näistä kullakin kategorialla on oma painoarvonsa, joka vaikuttaa lopullisissa yhteen lasketuissa pisteissä. Ohjelmia olisi voitu hyvinkin vertailla vielä laajemmalti useissa eri kategorioissa, mutta vertailuun otetut osa-alueet ovat opinnäytetyöprojektin kannalta kehittäjälle tärkeimmät. Muita osa-alueita kuvastamaan on vertailutaulukkoon otettu myös ”jokerikategoria” muut, joka pitää sisällään mm. verkko-ominaisuudet, kehittäjäyhteisön, moottorin suorituskyvyn sekä kehityskustannukset. Painoarvo kullekin kategorialle on nollan ja yhden välillä. Näin sillä voidaan laskea vähemmän tärkeiden osa-alueiden vaikutuksia kokonaistulokseen. Koska ”muut ominaisuudet” arvosteluosa-alue edustaa useaa eri kategoriata, on sille annettu suurin painoarvo. Pisteet kustakin kategoriasta annetaan yhden ja kymmenen väliltä. Vertailu on tehty taulukkoon 1.

Taulukko 1. Pelimoottorien vertailutulokset (Painoarvo merkitty sulkeisiin)

	Unity 3D	Unreal Engine 4*	Clickteam fusion 2.5	Game Maker: Studio
Käyttäjästävällisyys (0,8)	$8 \times 0,8 = 6,4$	$7 \times 0,8 = 5,6$	$9 \times 0,8 = 7,2$	$5 \times 0,8 = 4$
Graafinen suorituskkyky (0,6)	$8 \times 0,6 = 4,8$	$10 \times 0,6 = 6$	$6 \times 0,6 = 3,6$	$4 \times 0,6 = 2,4$
Ohjelmointikielet (0,6)	$10 \times 0,6 = 6$	$8 \times 0,6 = 4,8$	$7 \times 0,6 = 4,2$	$7 \times 0,6 = 4,2$
Muut ominaisuudet (1)	9	8	6	6
Yhteensä	26,2/30	24,4/30	21/30	16,6/30

Vertailussa parhaiten pärjännyt Unity 3D sai kärkisijansa sen käyttäjäystävällisen käyttöliittymänsä ansiosta. Sen graafinen suorituskkyky on vertailluista lähes samalla tasolla Unreal Engine 4:n kanssa, mutta se vaatii enemmän työtä saada yhtä näyttävää jälkeä kuin Unreal Enginellä. Unity sai parhaat pisteet muista ominaisuuksista juurikin sen kattavan asset storen ansiosta, joka tarjoaa kaikki puuttuvatkin ominaisuudet käyttäjien toteuttamana. Unityn kehittäjäyhteisö on kaikista vertailluista kehitysohjelmista suurin ja jatkaa yhä kasvamistaan. Unityn helppous ja halvat kustannukset on myös eduksi indie-kehittäjille.

Toiseksi vertailussa sijoittunut Unreal Engine 4 on markkinoilla olevista ilmaisista pelinkehitysovelluksista toinen, joka nauttii suuresta suosiosta. Sen sijoitus vertailussa on sen graafisten ominaisuuksien sekä suuren kehittäjäyhteisön ansiota. Muiden ominaisuuksien pisteistä kuitenkin laskevat hieman sen korkeat kustannukset kehittäjälle suhteessa Unity 3D:hen. Unitylle täytyy ostaa pro lisenssi vasta, kun 100 000 \$ vuosituotot täyttyvät. Pro lisenssi maksaa vain 1500 \$ kertamaksuna, tai 75 \$/kk, kun taas vastaavasti 100 000:n \$ tuotoilla joutuu kehittäjä

maksamaan Unreal Engine 4:n kehittäjille 5 % kokonais tuotosta, mikäli vuosineljänneksen tuotot ylittävät 3000 \$.

Kolmanneksi ja neljänneksi sijoittuneet Clickteam Fusion 2.5 sekä Game Maker: Studio joutuvat tyytymään huonompiin pisteisiin lähinnä niiden moottoreiden rajallisten graafisten kykyjen takia. Molempia ohjelmia myös vaivaa nykyään kuihtuva kehittäjäyhteisö, kun kehittäjät siirtyvät pääasiassa Unityyn tai Unreal Engineen. Clickteam Fusion 2.5 sai kuitenkin hyvät pisteet sen käyttäjäystävällisestä käyttöliittymästä ja helposti lähestyttävästä kehitysympäristöstä, kun taas Game Maker: Studio sai alhaiset pisteet juurikin samoista syistä.

Vaikkakin valinta pelimoottorin suhteen oli jo selkeä ennen tämän vertailun tekemistä, oli vertailu kuitenkin hyvä osoitus siitä, että valinta oli oikea. Tässä tulee tietysti ottaa huomioon, että Unreal Engine 4:ää ei päästy vertailua varten kokeilemaan, vaan sen suhteen vertailussa oltiin täysin käyttäjien antamien kokemusten varassa. Vaikka vertailu on yritetty toteuttaa mahdollisimman puolueettomasti, on Unity saattanut saada pisteitä helpommin kokemuksen ansiosta.

3 Peli-idea

Medieval Battlefield on kolmiulotteinen, isometrisesti renderöity, strategiapeli kahdelle pelaajalle verkon välityksellä. Pelin teema sijoittuu keskiaikaan, joskin joitakin vapautuksia pelin teemassa on otettu.

3.1 Pelin tavoite

Pelin tavoitteena on lyödä toisen pelaajan joukot. Tämä tapahtuu haastamalla toinen pelaaja taisteluun ”yleisnäkymässä”, jossa näkyy pelaajan taitotasoa vastaavia muita pelaajia. Tästä näkymästä pelaaja voi lähettää haasteen toiselle pelaajalle, joka saa puhelimeensa ilmoituksen saapuneesta haasteesta. Halutessaan haastettu pelaaja voi hyväksyä haasteen ja taistella itse haastajaa vastaan. Mikäli haastettava kuitenkin hylkää haasteen, pelaajan joukot ottaa haltuun tekoälypelaaja. Tällöin haastettu pelaaja ei saa taistelusta kokemuspisteitä.

Jos haastava pelaaja voittaa taistelun toista pelaajaa vastaan, hän saa siitä kokemuspisteitä. Kun kokemuspisteitä kertyy pelaajalle tarpeeksi, hän saa uuden kokemustason, joita keräilemällä pelaajan käyttöön avautuu uusia joukkotyypppejä, varusteita sekä kutsumuspisteitä. Pelissä on myös maailmanlaajuinen pistetilasto, jossa korkeimmalla tasolla olevat pelaajat voivat kilpailla globaalista herruudesta.

Näillä mekaniikoilla peli tarjoaa sekä kasuaalille satunnaispelaajalle mukavaa ajan vietettä. Sekä pelistä tosissaan innostuvalle pelaajalle sisältöä useiksi tunneiksi päivässä.

3.2 Pelin mekaniikat

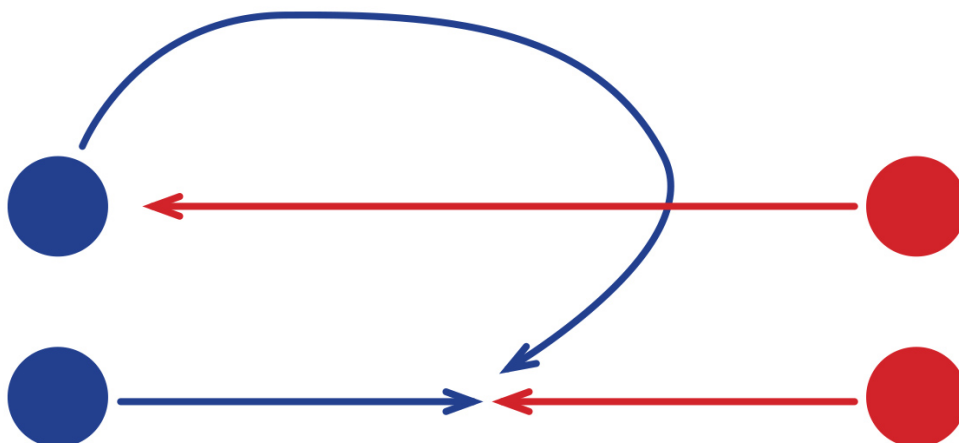
Medieval battlefieldissä on tarkoitus kasata oma armeija pelaajan käytössä olevista joukoista. Kun pelaaja on kasannut haluamansa armeijan, on tarkoitus hyökätä tai puolustaa vastustajan armeijaa, jota ohjaa toinen pelaaja tai tekoäly, vastaan

virtuaalisella taistelukentällä. Joukkojen liikuttelu tapahtuu vuoropohjaisesti kuitenkin siten, että molemmat pelaajat antavat käskynsä joukoilleen samanaikaisesti.

3.2.1 Taistelu

Taistelu alkaa, kun pelaaja päättää hyökätä vastustajan kimppuun. Tätä varten pelaajalla on käytössään lista, josta pelaaja voi katsoa muita pelaajia sekä heidän kokemustasoaan, ja valita itselleen sopivan. Peliin tulee myös mahdollisuus antaa pelin valita pelaajalle tasaväkisen vastustajan. Kun vastustaja on valittu, saa vastustaja ilmoituksen puhelimeensa, että hänellä on odottava haaste. Vastustaja voi halutessaan hyväksyä haasteen tai jättää taistelun tekoälyn ratkaistavaksi. Mikäli vastustaja ei ole paikalla, tekoäly ottaa automaattisesti vastustajan joukot käyttöönsä.

Kun taistelun molemmat osapuolet on valittu, valitsevat molemmat sopivan armeijan valmiiksi kasaamistaan armeijoistaan. Mikäli vastassa on tekoäly, se ottaa käyttöön satunnaisesti jonkin pelaajan armeijoista. Armeijat koostuvat noin 4-10 ennalta määritetystä ryhmästä. Joukkoja voidaan käskyttää ryhmäkohtaisesti. Ryhmät koostuvat 1-10 yksilöstä sen mukaan, minkä tyyppinen ryhmä on kysymyksessä. Taistelu käydään siten, että molemmat pelaajat (tai tekoäly) antavat kaikille joukoilleen sormieleillä halutut liikkeet, jotka pelaajan joukoilla on määrä toteuttaa. Kun kaikki käskyt on jaettu ja pelaaja on valmis, tai vuoron aikaraja tulee umpeen, käyvät kaikki joukot liikkumaan annettuja reittejä. Jos joukot kohtaavat liikkueessaan vastustajan joukot, ne alkavat automaattisesti taistelemaan keskenään. Kullakin ryhmällä on omat vahvuutensa ja heikkoutensa, ja ryhmien väliset taistelut ratkaistaan sen perusteella. Yhteen taisteluun voi osallistua useampi ryhmä, tästä esimerkki kuviossa 7.



Kuvio 7. Esimerkki taistelun ensimmäisestä kierroksesta

Molemmat pelaajat tekevät alemmilla ryhmillään suoran hyökkäyksen toisiaan kohtaan, joka johtaa taisteluun keskellä kenttää. Punainen pelaaja kuitenkin olettaa sinisen pelaajan tekevän saman liikkeen myös ylemmällä ryhmällä, mutta sininen pelaaja tekee kuitenkin pienen kiertoliikkeen yläpuolelta. Tämä johtaa siihen että kun käskyt toteutetaan, juoksee ylempi punainen suoraan ohi selustaan, ja ylempi sininen pääsee hyökkäämään meneillään olevaan taisteluun oman ryhmänsä avuksi, ja näin olleen varmasti kukistaa toisen punaisen pelaajan ryhmistä. Kun ensimmäinen kierros on käyty, on punaisella enää yksi ryhmä käytössään, ja sinisellä kaksi, jolloin sininen on ylilyöntiasemassa. Kierrosten pituus on myös rajattu ajalla, jolloin pelaaja ei voi käyttää useita minuutteja miettiessään sopivia hyökkäyksiä.

Taistelun vuoroja käydään niin kauan kun jommankumman kaikki joukot on lyöty, tai maksimi määrä kierroksia tulee täyteen, jolloin voittaja on se, jolla on voimakkaammat joukot elossa. Taistelun voittanut osapuoli saa kokemuspisteitä enemmän kuin hävinnyt. Mikäli puolustaja oli tekoälyn ohjaama, ei tämän joukot saa kokemuspisteitä.

3.2.2 Armeijat

Pelaajalla on käytössä tietty määrä ”kutsumus pisteitä”, joiden määrä riippuu siitä millä kokemustasolla pelaaja on. Näitä kutsumuspisteitä pelaaja voi käyttää kasatessaan itselleen armeijoita. Voidaan esimerkiksi toteuttaa niin, että kullakin kokemustasolla pelaaja saa käyttöönsä 5 kutsumuspistettä lisää. Yhden perus jalkaväkiryhmän armeijassa pitäminen maksaa 15-50 kutsumuspistettä, riippuen kuinka kehittyntä pelaajan jalkaväki on. Näin ollen esimerkiksi pelin alussa, jos pelaajalla on käytössään 100 kutsumuspistettä, voisi hän kasata armeijansa 5-6 jalkaväkiryhmästä.

3.2.3 Kasarmit ja väsyminen

Pelaajalla voi olla myös reservissä/kasarmilla ryhmiä, jotka hän ottaa käyttöön armeijaansa jos/kun joku toinen ryhmä on levon tarpeessa. Pelaajalla on kuitenkin myös reservilleen maksimi koko, noin 3 kertaa aktiivisen armeijan koko. Jos tämä tulee täyteen, täytyy pelaajan poistaa olemassa olevia joukkoja.

Kun joukoilla on taisteltu esimerkiksi 3-4 taistelua, on heidän jaksaminen heikentynyt sen verran että täytyy ryhmät laittaa lepäämään kasarmille, 4-12 tunniksi oikeaa aikaa. Pelaajalla voi olla yhtäaikaisesti käytössään useita ryhmiä, joiden välillä hän voi vaihdella, kun toiset ryhmät ovat lepäämässä. Kasarmeilla pelaajan joukkojen palautumista on nopeuttamassa parantajat. Näitä parantajia pelaajalla voi olla käytössään maksimissaan 4. Pelin premium-valuuttaa voidaan käyttää virkistysjuomien tai parantajien ostoon, joita käytetään kasarmilla lepäävien joukkojen nopeampaan palautumiseen. Näin ollen panostamalla premium-valuuttaan voi pelaaja käytännössä halutessaan taistella jatkuvasti.

3.2.4 Kokemustasot

Pelaajalle kertyy kokemuspisteitä kaikista taisteluissa, joissa pelaaja on itse ohjannut joukkojaan. Kokemuspisteitä kertyy enemmän pelaajalle, kun hän voittaa käydyn

taistelu. Kokemuspisteitä ei kuitenkaan vähennetä missään tilanteessa. Kokemustaso on eri asia kuin taitotaso.

3.2.5 Taitotaso

Pelaajan taitotaso kasvaa, tai laskee riippuen siitä, kuinka hyvin hän pärjää taisteluissa. Tällä pyritään saavuttamaan kaikille pelaajille mieluinen pelikokemus, jolloin pelaajan ei pitäisi joutua missään vaiheessa tilanteeseen, jossa hän ei enää pärjäisi vastustajille. Kun pelaaja voittaa itseään paremman pelaajan, nousee pelaajan taitotaso huomattavasti, kun taas pelaaja häviää itseään paremmalle, ei tappiolla ole vaikutusta pelaajan taitotasoon. Hävitessään, tai voittaessaan saman tasoiselle pelaajalle, laskee, tai nousee pelaajan taitotaso vastaavasti. Huonompaa pelaajaa vastaan voittaessa, pelaajan taitotaso ei sen sijaan nouse. Mutta mikäli pelaaja häviää huonommalla taitotasolla olevalle pelaajalle, hänen taitotasonsa laskee huomattavasti.

3.3 Joukkotyypit

3.3.1 Yleistä

Luvussa 3.3 esitellyistä joukkotyypeistä vain footman on pelin opinnäytetyön versiossa käytettävissä. Muut joukkotyypit toteutetaan kesän 2015 aikana pelin julkaisuversiota varten.

Jokaisesta käydystä taistelusta pelaajalle ja pelaajan joukoille kertyy kokemuspisteitä. Kokemuspisteitä voidaan ”mikromanageroida” joukoilla, tai pelaaja voi valita automaattisen kehityksen joukoilleen. Tällöin tekoäly huolehtii pelaajan joukkojen kehittämisestä. Pelaajan vastuulle jää kuitenkin vielä joukkojen hallinta armeijatasolla. Näin peli tarjoaa paljon säätämistä sitä haluaville pelaajille, mutta myöskin kevyen taukopelin kasuaalimmalle pelaajalle.

3.3.2 Footman

Footman on tavanomainen miekkamies. Footmanin vahvuus on sen yleispätevyys, se kykenee liikkumaan keskipitkiä matkoja ja tekee vahinkoa muihin joukkotyyppeihin verrattaen keskiverrosti. Footman myös kestää hyvin iskuja. Footman-ryhmä voidaan aseistaa joko miekalla ja kilvellä, jolloin ne kestävät hieman paremmin iskuja, mutta myös tekevät vähemmän vahinkoa, kahdella miekalla jolloin ne tekevät enemmän vahinkoa, mutta eivät kestä niin paljon tai seipäällä, jolloin ne eivät kestä niin paljon iskuja, mutta pystyvät hyökkäämään ennen muita ryhmätyyppejä lähietäisyydeltä. Seipään etu on myös tavallista footmania nopeampi ja pidempi liikkumamatka. Kuviossa 8 on footman, jolla on varusteinaan miekka ja kilpi.



Kuvio 8. Footman varustettuna miekalla ja kilvellä

3.3.3 Archer

Archerin erikoisuus on sen kyky ampua nuolia. Tästä johtuen archer on hyvä tukijoukko footmaneille sekä knighteille. Archer ei itse kestä juurikaan iskuja, joten se on hyvä pitää suojassa vihollisen jalkaväeltä. Se on kuitenkin nopea jaloistaan, ja pystyy pakenemaan jahtaavia knighteja ja footmanuja, ainoana poikkeuksena seipäällä varustettu footman. Archer ei pysty kuitenkaan samalla vuorolla liikkumaan sekä ampumaan, joten yksin archer ryhmä ei pärjää jahtaavia jalkajoukkoja vastaan. Archer voidaan aseistaa joko isolla kaarijousella, joka tekee enemmän vahinkoa ja kantaa pidemmälle, tai vaihtoehtoisesti jalkajousella, jonka osumaetäisyys on lyhyempi, mutta jalkajousi voidaan ampua useamman kerran vuorolla.

3.3.4 Knight

Knight on jokaisen komentajan armeijan tukipilari. Se kestää paljon iskuja, ja tekee paljon vahinkoa, kääntöpuolena tälle on se että knight ei ole kovin nopea, eikä pysty juoksemaan pitkää matkaa vuorollaan. Knightilla ei varsinaisesti muuta heikkoutta ole kuin sen nopeus. Knight voidaan kuitenkin varustaa hevosella, jolloin knightista tulee myös huomattavasti nopeampi joukko. Tällöin sillä on kuitenkin aseenaan vain yhden käden miekka, jolla se ei pysty niin suureen vahinkoon. Omin jaloin liikkuvalla knightilla on käytössään joko kahden käden miekka, tai suuri sotakirves. Sotakirves tekee eniten vahinkoa, mutta tekee knightista hitaamman taistelussa, jolloin knight on koko sotajoukon hitain lyöjä.

4 Peli-ideasta peliksi

Tässä luvussa käydään läpi mitä tulee ottaa huomioon lähtiessä yksin, tai pienellä porukalla, kehittämään peliä.

4.1 Pelinteko prosessina

4.1.1 Suunnittelu

Moni indiepeli saa alkunsa halusta toteuttaa mielestään hyvä peli-idea. Ideat lähtevät usein liikkeelle jostain kehittäjälle mieleisesti pelistä tai peligenrestä, johon halutaan tuoda jokin oma uusi ”twisti”. Osa ideoista taas on lähes samoja kuin jo olemassa olevat pelit, mutta kehittäjä haluaa toteuttaa vastaavan pelin paremmin tai haluaa vain lisää kokemusta pelien kehittämisestä. Myös opinnäytetyön projekti sai alkunsa tällä tavalla. Se lähti halusta tehdä yksinkertainen startegiapeli keskiaikaisella teemalla, johon piti jokin pieni uusi idea tuoda. Pelin yhtäaikainen käskyjen jakaminen sai ideana pohjansa Game of thrones -lautapelistä, jossa jokainen pelaaja antaa omille alueilleen yhtä aikaan käskyn, jotka paljastetaan kun kaikki joukot ovat saaneet käskynsä. Myös tietokoneelle julkaistu Endless Legend -peli antoi oman osansa ideoista. Se on ruudukkopohjainen 4X-peli, jossa taistelevat osapuolet antavat vuorotellen kaikille omille joukoilleen käskyt.

Kun peli-idea on jokseenkin selkeytynyt kehittäjän mielessä, kannattaa lähteä kirjoittamaan ideoita puhtaaksi paperille esimerkiksi ranskalaisin viivoin tai täysin kappalein. Kun ideat on paperilla kannattaa ne lukea läpi. Näin yleensä huomaa jos jokin ei toimi ja jos sitä kannattaa muuttaa tai vaihtoehtoisesti jos on jotain olennaista unohtanut. Peli-idea kirjoittaessa on hyvä miettiä läpi mitä peliprojektilla yritetään saavuttaa, ja mikä on itse pelin tavoite. Projektin voi aloittaa esimerkiksi omaksi vapaa-ajan puuhaksi, kehittäjälle taloudelliseksi hyödyksi tai kuten tämän opinnäytetyöprojektin tapauksessa, tavoitteena on tehdä valmis peli referenssiksi

projektin toteuttajalle tulevaisuuden työnhakua varten, sekä sivussa tienata muutama euro mainostuloista. Peli-ideaa miettiessä on myös otettava huomioon onko se mahdollista toteuttaa järkevässä ajassa käytössä olevilla resursseilla.

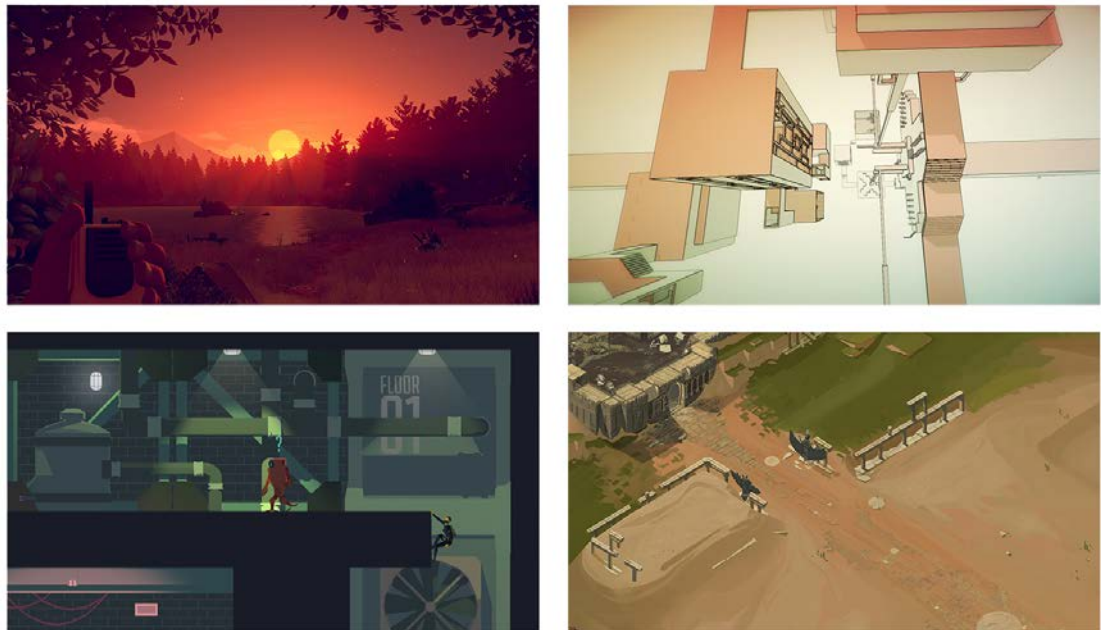
Jos ja kun peli-idea tuntuu järkevältä toteuttaa, on hyvä lähteä etsimään vastaavia pelejä. Näitä pelejä on hyvä myös kokeilla, koska niistä voi saada hyviä ideoita omaan projektiinsa, tai vastaavasti niistä voi oppia mitä kannattaisi toteuttaa toisin. On myös suositeltavaa etsiä, mikäli löytyisi vastaava peliprojekti, josta on dokumentoitu itse projektin kehitys. Siitä voi käydä ilmi mahdolliset ongelmakohdat vastaavassa projektissa. Näin pystyy ottamaan jo alun suunnittelussa huomioon mahdolliset ongelmat. Opinnäytetyön projektia suunniteltaessa ei juurikaan etsimällä etsitty vastaavia projekteja, vaan inspiraatio ja ideat löytyvät peleistä, joita kehittäjä oli jo aiemmin kokeillut.

Kun idea on selkeytynyt ja kirjoitettu paperille ylös, on hyvä miettiä onko peli järkevää tai edes mahdollista toteuttaa yksin. Tätä voi lähteä purkamaan miettimällä ensiksi mitä itse osaa tehdä sekä mitä itse haluaa tehdä. Jos kehittäjältä itseltään löytyy kaikki tarvittavat taidot pelin kehittämiseen, on siltin hyvä miettiä, että riittääkö kiinnostus tai aika kuitenkin kaikkeen vaadittavaan työhön. Jos kehittäjä tarvitsee pelinsä toteuttamiseen muita henkilöitä avuksi on syytä ottaa huomioon, ettei aikataulut tule pitämään sitäkään vähää pelin kehittämisessä mikäli kehittäjä ei ole valmis maksamaan palkkaa ulkopuolisille henkilöille. Kuten eräässä mainoksessa sanotaan: aikataulu ja budjetti kerrotaan kolmella.

Peliä suunnitellessa on myös hyvä pitää mielessä alusta asti mille alustalle pelin aikoo julkaista. Tämä vaikuttaa muun muassa siihen, millaiset grafiikat peliin voi toteuttaa, kuinka paljon laskentatehoa on käytössä monimutkaisiin laskuihin ja mitä syöttölaitteita on käytettävissä. On mahdottomuus esimerkiksi pelata mobiililaitteella peliä, joka on suunniteltu alun perin pelattavaksi tietokoneen näppäimistöllä, ellei pelin syöttömetodeja suunnitella täysin alusta. Opinnäytetyön

projekti suunnitellaan ja toteutetaan alusta alkaen mobiililaitteelle, mutta sormieleet mahdollisesti vaihdetaan hiirieleiksi mikäli peli julkaistaan myös tietokoneelle.

Jo suunnitteluvaiheessa on hyvä myös miettiä millaisen graafisen ilmeen pelille antaa. Nykyään indiekehittäjillä on tapana antaa peleilleen yksinkertainen visuaalinen ilme. Tämä on ymmärrettävää, koska yksinkertaista grafiikkaa on nopea tuottaa ja siten saa myös helposti näyttävää jälkeä aikaan. Yksinkertainen visuaalinen ilme on yleensä myös kevyempää renderöidä, joten se sopii hyvin myös mobiililustoille kehittäessä. Kuvioon 9 on koottu esimerkkejä erilaisista yksinkertaisista visuaalisista ilmeistä. Graafisessa suuntauksessa kannattaa myös miettiä onko pelin tarpeen olla 3-ulotteinen, vai riittääkö pelkästään 2-ulotteinen grafiikka pelille. 2-ulotteinen grafiikka on kuitenkin yleensä nopeampaa tuottaa, mikäli animoitavaa ei ole paljon.



Kuvio 9. Esimerkkejä yksinkertaisesta graafisesta tyylistä

4.1.2 Toteuttaminen

Kun peli-idea on valmis, mahdolliset sudenkuopat on etukäteen mietitty, graafinen tyyli linjattu, kehitystiimi kasattu ja aikataulu suunniteltu, on aika aloittaa pelin

toteuttaminen. Pelin kehitysvaiheessa on tärkeintä pysyä itselleen asettamissaan aikatauluissa, jotta pelin kehitys ei pääsisi hidastumaan tai mielenkiinto laskemaan.

Jos peli on suunniteltu hyvin, ei toteuttamisvaiheessa pitäisi olla suurempia ongelmia vastassa. Jos kuitenkin kehittäjä törmää ongelmiin pelin kehityksessä, on todennäköisimpänä syynä kehittäjälle entuudestaan tuntematon tekniikka. Kuten hyvin todennäköistä, etenkin kokemattomalle kehittäjälle, jossain vaiheessa projektia aikataulut lipsuvat. Kun näin käy, kannattaa kehittäjän lähteä miettimään läpi projektiaan ja sitä onko siinä jotakin, josta karsia pysyäkseen aikataulussa. Esimerkiksi hienot siirtymäanimaatiot eri valikoiden välillä tai pelihahmojen vaatteiden fysiikat ei välttämättä pelin pelattavuuden ja viihdyttävyyden kannalta ole tärkeimpiä. Jos kehittäjä ei kuitenkaan halua karsia suunnitelmiaan ominaisuuksia pelistä pois, on varauduttava aikataulun venymiseen. Se ei kuitenkaan pidempään kehittäessä ole kaikkein kannattavinta, koska usein kehittäjän innostus pelin kehitykseen laskee, eikä pelin teosta enää nauti, vaan siitä tulee pakkopullaa. Hyvänä esimerkkinä tällaisesta projektista voidaan ottaa overgrowth. Sen kehitys on aloitettu jo vuonna 2008, ja peli on yhä alpha vaiheessa. Peliin kyllä tulee yhä tasaiseen tahtiin päivityksiä, mutta päivitykset sisältävät lähinnä parannuksia hahmojen animaatioiden fysiikoihin tai pikselin tarkkojen haavoja/verenvuodatus-efektejä.

Yksi parhaita puolia yksin peliä kehittäessä on se, että pääsee itse päättämään järjestyksen, jossa pelin eri osa-alueita kehittää. Tästä on hyötyä esimerkiksi tilanteessa, jossa ei saa jotain ominaisuutta pelimoottorista toimimaan, niin voi kehittäjä siirtyä hetkeksi tekemään esimerkiksi grafiikkaa peliinsä. Näin mieli hieman virkistyy ongelmakohdan osalta, ja jopa muutama tuntikin muuta puuhaa saattaa auttaa keksimään ratkaisun ongelmaan. Totta kai myös tässä on oltava varovainen, jotta ei liikaa hypi eri tekemisten välillä, jotta ei kehittäminen tyssää siihen kun kehittäjä on tehnyt peliään useita kuukausia, mutta kasassa on vasta pirstaleita moottorista, grafiikasta, musiikista ja 3D-malleista.

4.1.3 Julkaisu

Kun peli on vihdoin valmis julkaistavaksi, on pelille jo varmasti mietitty julkaisualusta. Tästä syystä onkin tärkeää jo pelin kehitysvaiheessa ottaa huomioon mitä eri julkaisualustat vaativat kehittäjiltään, jotta ikäviltä yllätyksiltä välttyään.

Mikäli peli on toteutettu pienellä budjetilla, haluaa kehittäjä todennäköisesti myös julkaista pelinsä kustannustehokkaasti. Tästä syystä Google Play sekä Steam greenlight ovat hyviä lähestymissuuntia julkaisulle. Molemmat ovat ilmaisia kehittäjälle. Steam Greenlight vaatii kuitenkin pelaajien hyväksynnän, jotta peli voidaan julkaista Steamin kauppapaikalle. Jos kehittäjällä on kuitenkin pieni summa sijoitettavana julkaisua varten, ei julkaisuoikeus iOS:n mobiilikauppaankaan ole kovin kallista, maksaessaan 99 \$ vuodessa.

Julkaisun jälkeen kehittäjän kannattaa hetki seurata pelinsä vastaanottoa. Mikäli vastaanotto pelille on hyvä, on kehittäjän syytä harkita pelinsä tuen jatkamista. Se voi olla esimerkiksi vain bugikorjailuja tai vaihtoehtoisesti täysiä lisäosia. Lisäosien julkaisulla kehittäjä voi tavoitella itselleen lisää rahaa. Toinen mahdollisuus on julkaista ilmaisia lisäosia, tämä on kannattavaa silloin, kun huomaa pelinsä suosion laskevan tai se ei ole koskaan noussutkaan kunnolla. Näin saadaan vanhat pelaajat houkuteltua pelin pariin takaisin sekä uudet pelaajat kiinnostumaan pelistä, koska usealle pelaajalle on positiivinen merkki jos pelin kehitys on aktiivista.

4.2 Testausmenetelmät mobiilipelissä

Testausmenetelmät mobiilipelissä on pitkälti samat kuin mitä tahansa mobiilisovellusta testatessa. Se koostuu black-box testauksesta sekä white-box testauksesta. Black-box testauksessa testataan mm. graafista käyttöliittymää, käyttäjäkokemusta sekä muita visuaalisia elementtejä pelissä. White-box testauksessa taas kiinnitetään enemmän huomiota sovelluksen arkkitehtuuriin,

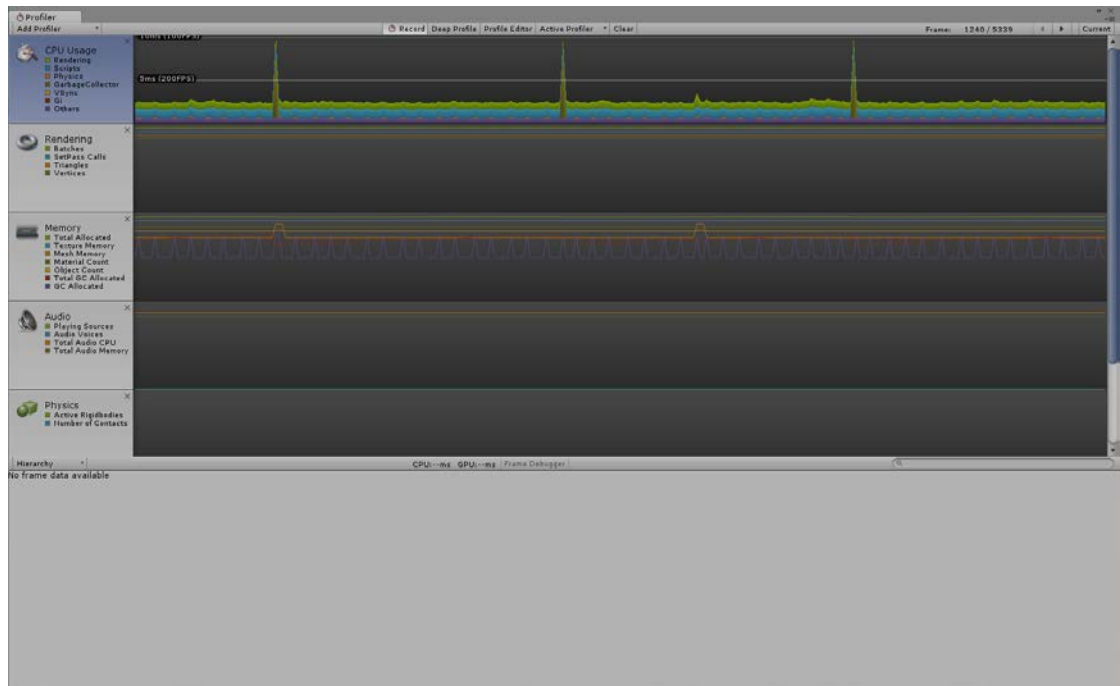
siihen kuinka hyvin pelissä käytetyt eri komponentit toimivat keskenään, kuten grafiikka- ja pelimoottori, kolmannen osapuolen komponentit jne. (Ville-Veikko 2014)

Peliprojektin testausmetodit ovat toiminnallisuuden testaus, yhteensopivuuden testaus, suorituskyvyn testaus, kuormituksen testaus (sekä lokalisaation testaus).

Toiminnallisuuden testaus on yleensä mobiilipeliä testatessa manuaalista pelin läpi koluamista. Suuremmissa projekteissa aikataulussa pysymisen sekä ketterien kehitysmenetelmien kannalta on kuitenkin tärkeää toiminnallisuuden testauksen automatisointi. Käytännössä se tarkoittaa jonkin valmiin frameworkin tuomista omaan projektiin, joka testaa virtuaalisesti pelin sisältöä läpi etsien mahdollisia ongelmakohtia esimerkiksi pelin stabiiliudessa tai pelin mekaniikoissa. Medieval battlefieldin testauksessa käytettiin ainoastaan manuaalista pelaamista, jolla mahdolliset ongelmakohdat pelistä pyrittiin löytämään.

Yhteensopivuuden testauksella mobiilipeleissä tarkoitetaan pelin testausta eri alustoilla. Tämä on hyvä suorittaa useita kertoja pelin valmistuessa, jotta kehittäjälle ei tule yllätyksenä mahdolliset ongelmat eri laitealustoilla. Medieval battlefieldin tapauksessa yhteensopivuuden testausta kehityksen aikana ei käytännössä ollut lainkaan. Tämä johtuu pelin pienestä budjetista, joka esti suurien laitehankintojen tekemisen.

Suorituskyvyn testaus on tärkeä osa mobiilipelin kehityksessä, se kulkee rinta rinnan yhteensopivuuden testauksen kanssa. Suorituskykyä testatessa on yleensä apuna kehittäjällä kehitysovelluksen omat diagnostiikkatyökalut joilla kehittäjä voi seurata usein hyvin yksityiskohtaisestikin mikäli ohjelmakoodissa on pullonkauloja. Esimerkiksi Unity 3D:stä löytyy profiler-työkalu (kuvio 10), joka antaa tarkkaa tietoa esimerkiksi kuinka paljon mikäkin funktio käyttää prosessorista, ja mihin sovellus käyttää muistia.



Kuvio 10. Unity profiler

Kuormituksen testauksella testataan pelin toimintaa kuorman alla, se voi tarkoittaa esimerkiksi pelin palvelimien kuormittamista useilla yhtäaikaisilla pelaajilla tai graafisen suorituskyvyn kuormittamista piirtämällä useita tuhansia objekteja ruudulle. Kuormituksen testaaminen on hyödyllistä muun muassa siksi, että se tarjoaa kehittäjälle tärkeää tietoa pelin suorituskyvystä sekä moottorin rajoitteista. Medieval battlefieldin kannalta kuormituksen testaaminen on tärkeää, jotta voidaan todeta esimerkiksi kuinka monta yksikköä, ja yksiköiden välistä taistelua voidaan ruudulle yhtä aikaa piirtää.

Lokalisaation testaus on tärkeää silloin, kun peli tähtää kansainvälisille markkinoille. Sillä tarkoitetaan käytännössä pelin eri kieliversioiden oikolukemista.

Kuusi avainkohtaa pelinkehityksen testausmetodeissa (mukaillen Ville-Veikko 2014):

Käyttöliittymä ja sen toiminnallisuus

Käyttöliittymä ja sen toiminnallisuuden testaus on yksi tärkeimmistä osa-alueista pelin testauksessa. Sillä voi olla suurikin vaikutus pelin suosioon. Jos peli toimii vain murto-osalla eri alustoista hyvin, ei se yksinkertaisesti tavoita kaikkia mahdollisia asiakkaitaan. Tästä syystä onkin tärkeää kokeilla käyttöliittymän toimintaa useilla eri laitealustoilla, sekä eri suuruisilla näytöillä. On myös hyvä ottaa huomioon onko pelin tarkoitus toimia esimerkiksi vain mobiililaitteen ollessa vaakasuorassa, vai tukeeko käyttöliittymä molempia orientaatioita.

Graafinen suorituskyky

Jotta pelin graafinen suorituskyky olisi tasainen useiden eri laitealustojen välillä, kannattaa peliä koeajaa mahdollisimman monella eri laitteella. Tämä on erityisen tärkeää peleissä, joissa useampi pelaaja kilpailee reaaliaikaisesti toisiaan vastaan. Toisella huonosti pyörivä sovellus antaa toiselle pelaajalle etulyöntiaseman, mikä taas vaikuttaa negatiivisesti toisen pelaajan kokemukseen.

Käytettävyys ja käyttökokemus

Mobiilipeliä testatessa käytettävyyden testaus on syytä hoitaa suoraan mobiililaitteella, hiirellä ja näppäimistöllä ei kehittäjäkään pääse kunnolla kokeilemaan pelinsä käyttökokemusta.

Moninpeliominaisuudet

Moninpeliominaisuudet on erityisen tärkeää testata mikäli pelin paino on nimenomaan verkkopelissä. Mikäli pelin palvelimet kaatuilevat ja yhteys käyttäjän laitteeseen katkeaa jatkuvasti, vaikuttaa se erittäin negatiivisesti käyttäjän kokemukseen pelistä, ja pahimmillaan jopa saa käyttäjän hylkäämään pelin.

Sosiaaliset ominaisuudet

Pelistä riippuen, voi sosiaaliset ominaisuudet olla hyvinkin tärkeitä pelisi menestyksen kannalta. On kuitenkin hyvää ilmaista markkinointia pelille, jos peli sisältää esimerkiksi huippupisteiden jaon Facebookissa. Sosiaaliset ominaisuudet on syytä testata huolella useilla eri alustoilla kaikkiin pelille valittuihin medioihin.

Tietoturva

On varmasti sanomattakin selvää että nykyaikana on äärimmäisen tärkeää pelin tietoturva, niin käyttäjien kuin kehittäjänkin kannalta. Luo pelille ja kehittäjälle erittäin huonoa mainetta jos käyttäjän luovuttamat henkilökohtaiset tiedot vuotavat pelistä väärin käsiin. Tietoturvan testaaminen indie-kehittäjälle voi olla haasteellista, sillä ilman omaa kokemusta mahdollisten tietoturva-aukkojen löytäminen omasta koodista on äärimmäisen haastavaa. Suuremmat yritykset usein palkkaavat kolmansia osapuolia käymään sovellustaan läpi mahdollisten tietoturva-aukkojen varalta.

4.3 Grafiikan toteuttaminen peliin

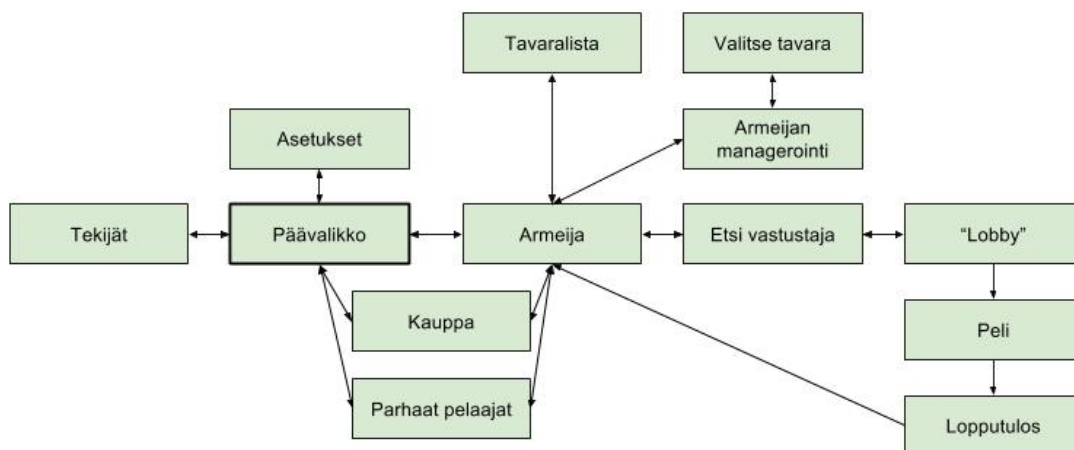
4.3.1 Graafisen käyttöliittymän ja käyttöliittymäkokemuksen toteuttaminen

Pelin menestyksen kannalta on hyvin tärkeää, että pelin käyttöliittymä on nopeasti omaksuttava ja hyvin toimiva. Käyttöliittymän tulee tarjota käyttäjälle suoraviivaisen ja helpon tavan käyttää peliä tai ohjelmaa. Jos käyttöliittymä on huonosti suunniteltu, liian hankala lukea tai yleisesti vain sekava, johtaa se huonoon käyttökokemukseen. Huono käyttökokemus taas johtaa pelin huonoon menestykseen ja sitä kautta pelin tuottavuuteen. Mikäli kehittäjällä on myös graafista osaamista kannattaa myös käyttöliittymän ulkoasuun panostaa. Vaikka grafiikalla ei ole niinkään väliä käyttökokemuksen kannalta, antaa se kuitenkin käyttäjälle viimeistellyn vaikutelman pelistä. Näistä syistä käyttöliittymän suunnittelu on yksi tärkeimpiä asioita pelin suunnittelussa ja toteuttamisessa.

Käyttöliittymän suunnittelu on hyvä aloittaa tutkimalla muiden mobiilipelien käyttöliittymiä ja niistä kannattaa pistää mieleen mikä toimii ja mikä ei. Muiden käyttöliittymiä ei kannata kuitenkaan lähteä suoraan imitoimaan, vaan kannattaa miettiä tarkkaan mitkä ovat oman pelin kannalta tärkeitä ja olennaisia ominaisuuksia ja panostaa niihin. Koska mobiilipelien käytössä oleva ruututila on usein varsin pieni, on käytettävyyden kannalta parempi että yhdellä näkymällä ei tuoda liikaa tietoa käyttäjälle. Sen sijaan on järkevämpi jakaa näytettävä tieto ja interaktio usealle eri näkymälle. Tällöin on pidettävä erityisen tarkkaa huolta siitä, että käyttäjä on jatkuvasti kartalla missä näkymässä hän sillä hetkellä on, sekä mistä/miksi hän siihen on päätenyt. (Derk 2012)

Kun kehittäjällä on jo hieman ajatuksia siitä mikä voisi toimia ja mitä tulisi välttää, tulisi miettiä tarkkaan läpi mitä eri ikkunoita pelin käyttöliittymä tarvitsee. Työkalulla, jolla käyttöliittymäsuunnitelma tehdään ei ole väliä, vaan kyse on enemmän kehittäjän omasta mieltymyksestä. Tämän voi tehdä helposti esimerkiksi Microsoftin Visiolla tai Google Driven piirtotyökalulla, myös perinteiset ranskalaiset viivatkin ovat varmasti avuksi kun käyttöliittymäsuunnitelmia lähdetään toteuttamaan.

Käyttöliittymän sujuvuus tulee jo tässä vaiheessa hyvin selville kehittäjälle, samalla se auttaa kehittäjää miettimään mitä eri nappeja tai navigointia kuvastavia elementtejä eri näkymiin tarvitaan. Kuviossa 11 on esimerkkinä suunnitelma Medieval Battlefieldin käyttöliittymän juoksusta, suunnitelma on toteutettu Google Driven piirtotyökalulla.

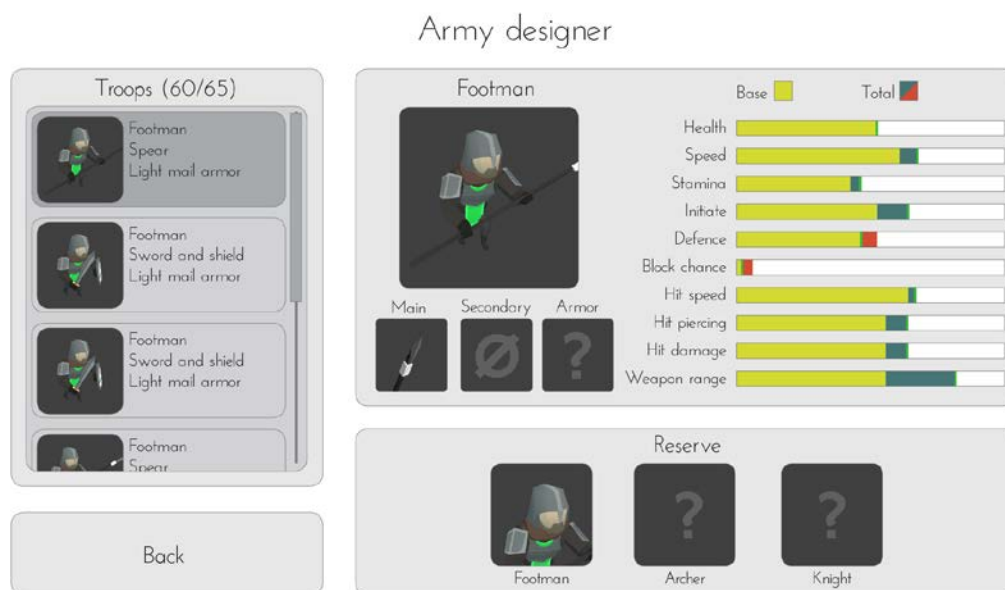


Kuvio 11. Suunniteltu hierarkia eri näkymien välillä

Kun pelin eri näkymät on suunniteltu, voidaan siirtyä miettimään yksittäisiä näkymiä. Tämän voi esimerkiksi hahmotella postit lapuilla, kynällä ja paperilla tai käytännössä millä tahansa työkalulla kehittäjän on asia helppo hoitaa. Tärkeintä tässä vaiheessa on hahmotella se mitä mikin ikkuna sisältää, eikä niinkään yksittäisten elementtien paikkoja. On myös hyvä pitää koko suunnittelun ajan mielessä kuinka paljon tietoa haluaa yhdellä ruudulla näyttää. Liika informaatio voi johtaa joidenkin käyttäjien kohdalla koko ruudun ohittamiseen, jolloin heiltä jää osa pelikokemuksesta kokematta. Jos vaikuttaa että yksittäiseen ikkunaan alkaa kertymään liikaa informaatiota tai interaktiivisia elementtejä, on järkevää lähteä miettimään voisiko osan sen ruudun sisällöstä siirtää muihin ruutuihin vai onko peräti tarpeen tehdä kokonaan uusi ruutu.

Kun näkymän sisältö on selkeytynyt kehittäjälle, voidaan näkymän asettelu suunnitella, eli tehdään niin kutsuttu rautalankamalli. Tätä suunnitellessa on huomioitava laitteet, joille peli on suunnattu, sekä onko valikon toimittava mobiililaitteen ollessa vaakana sekä pysty asennossa, vai riittääkö vain toisessa näistä. Yksi hyvä tapa testata käyttöliittymän toimivuutta tässä vaiheessa on tallentaa rautalankamalli kuvatiedostoksi ja siirtää se suoraan omaan mobiililaitteeseen. Silloin selviää yleensä ongelmakohdat mahdollisista liian pienistä elementeistä, joihin

käyttäjän on hankala osua sormillaan tai liian suuresta informaation määrästä. On syytä myös pitää jatkuvasti mielessä kuinka helppoja erilaiset käyttöliittymäelementit on toteuttaa ohjelmoimalla, jotta näiden ongelmien kanssa ei tarvitse pohtia kun käyttöliittymää lähdetään ohjelmoimaan. Kuviossa 12 on opinnäytetyöprojektin armeijan managerointi-ruutu hahmoteltuna InDesignissä. Työkalu tähän tehtävään valittiin täysin kehittäjän runsaasta kokemuksesta Adoben taitto-ohjelmien parissa.



Kuvio 12. Armeijan suunnittelunäkymän luonnos

Kun rautalankamallit on suunniteltu eri näkymille voidaan näitä näkymiä apuna käyttäen suunnitella näkymille visuaalinen ilme. Tässä vaiheessa voi päästää kehittäjä luovuutensa valloilleen. On kuitenkin hyvä myös tässä vaiheessa useasti kokeilla käyttöliittymää mobiililaitteella. Mobiililaitteiden näyttöjen pikselitiheydet ovat kuitenkin nykypäivänä huomattavasti korkeampia kuin työpöytäkoneiden. Tästä syystä pienet yksityiskohdat, jotka näyttävät kuvankäsittelyohjelmassa hyvältä voi näyttää mobiililaitteella täysin kelvottomilta. Mikäli kehittäjä osaa työstää vektorigrafiikkaa, kannattaa ulkoasusuunnitelmat ehdottomasti tehdä sillä. Tällöin on pelin käyttöliittymä helppo päivittää uusien laitteiden tarjoamiin suuriin

resoluutioihin ja pikselitiheyksiin. Kuviossa 13 on ulkoasuunonnos rautalankamallin pohjalta Adobe Photoshopilla toteutettuna.



Kuvio 13. Armeijan suunnitteluikkunan lopullinen grafiikka

Kuvion 13 ulkoasusuunnitelma on informaatiomäärältään ehkä jopa liian suuri, tästä johtuen peliin toteutettiin ”vinkkijärjestelmä”, joka osoittaa käyttäjälle käyttöliittymän eri kohtia ja kertoo, mikä niiden funktio on.

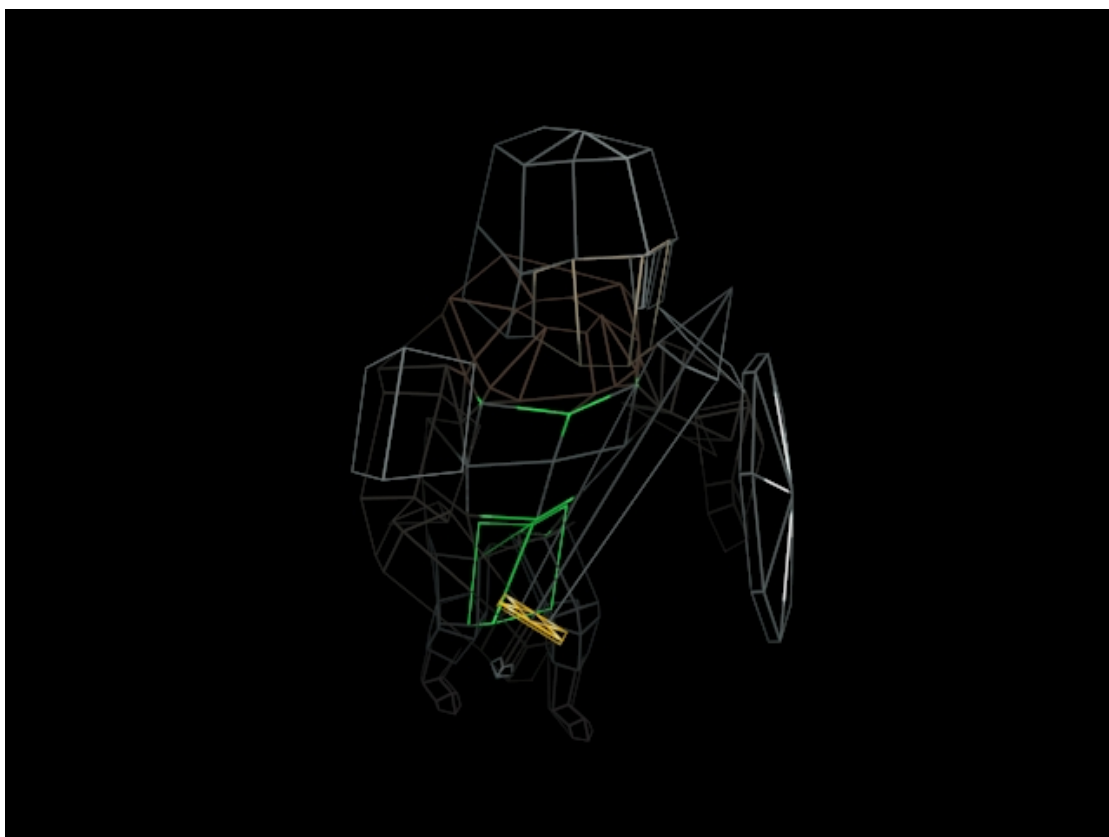
4.3.2 3D mallien mallintaminen

Kun peliin lähdetään miettimään 3-ulotteisen grafiikan käyttöä, on heti alkuun otettava huomioon mille alustalle kyseistä peliä tehdään. 3D-mallien toteuttaminen peleihin on poikkeavaa verrattuna elokuvaan tehtäessä, sillä pelin on pyörittävä tasaisesti, eikä suuri polygon määrä saa vaikuttaa negatiivisesti pelin toimintaan. Vaikka mobiililaitteidenkin rauta paranee jatkuvasti, on niidenkin kapasiteetit yhä rajallisia. Tästä syystä onkin yhä hyvä miettiä 2-ulotteisen grafiikan käyttämistä, etenkin jos pelissä on paljon objekteja yhtä aikaa ruudulla. Jos kehittäjä kuitenkin haluaa käyttää pelissään 3-ulotteista grafiikkaa, on jatkuvasti pidettävä mielessä ettei mobiililustoilla ole kuitenkaan rajattomasti kapasiteettia näyttää yksityiskohtaisia

3D malleja. Nykyaikaisilla tietokoneilla ja konsoleilla ei enää juurikaan ole rajoitteita 3D mallien polygoni määrissä, ainakaan hahmojen ja muiden tärkeiden objektien kohdalla.

Jotta mobiilipelien 3D malleihin saadaan lisää yksityiskohtia, on kehittäjällä mahdollisuus käyttää malleissaan materiaaleja, joilla saadaan luotua lisää yksityiskohtia käytettyihin malleihin. Materiaaleilla voidaan tuoda esimerkiksi pintaan kohonneita kohtia bump-kartalla, normal-kartalla tai tesselaatiolla. Itse pelin puolella materiaalit vaativat shaderin.

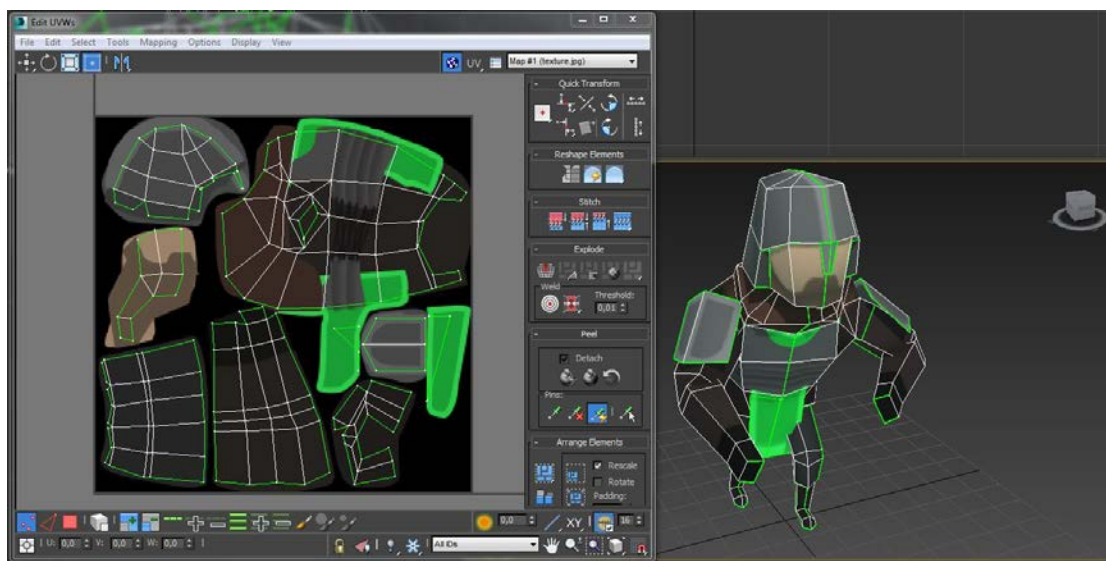
3-ulotteisen mallin mallintaminen aloitetaan miettimällä mihin tarkoitukseen kyseinen malli tulee. Jos kyseessä on esimerkiksi pelin päähahmo, voidaan sille varata huomattavasti suurempi määrä polygoneja kuin esimerkiksi kivelle, joka näkyy ruudun alalaidassa kesken taistelun. Vaikka kyseessä olisikin pelin osalta tärkeä malli, on myös silloin mietittävä tarkkaan mitkä kohdat mallista on tärkeitä ja käyttää niihin kohtiin enemmän polygoneja. Välttämättä ei kannata esimerkiksi hahmolle lähteä mallintamaan jokaista sormea erikseen, varsinkaan jos hahmolla on jatkuvasti hansikkaat kädessä. Mitään nyrkkisääntöä mallien polygon määristä ei ole, vaan mallien yksityiskotaisuus on mietittävä tapauskohtaisesti riippuen mallin käytöstä, kuinka läheltä mallia katsotaan sekä mille alustalle peli ollaan julkaisemassa. Kuviossa 14 on pelin footman-hahmosta rautalankarenderi. Malli on toteutettu 644 trianglella, eli kolmikulmaisella polygonilla. Malli on jätetty tarkoituksella mahdollisimman kevyeksi, koska hahmoja tulee olemaan useita kymmeniä pelikentällä yhtäaikaan ja hahmoa ei katsota kuitenkaan kovin läheltä. Tärkeintä mallissa on että sen erottuu muista pelin hahmoluokista, myös sen ollessa kaukana.



Kuvio 14. Rautalankamallinnos pelin footman-hahmosta

Hyvin optimoidun mallin mallintamisen jälkeen mallille täytyy tehdä pinta, eli se täytyy teksturoida. Tämä alkaa UVW-mappaamisella. Siinä mallin kaikki polygonit ikään kuin levitetään tasaiselle pinnalle. Polygonit kannattaa asetella niin, että samaa materiaalia olevat osiot ovat vierekkäin, esimerkiksi kypärän polygonit omana ryppäänään, housut omanaan ja ylävartalo omanaan. Tämäkin totta kai riippuu pitkälti siitä millainen malli on kyseessä. Kun kaikki on polygonit on on aseteltu järkevästi, voidaan UVW-mapista renderöidä kuvatiedosto, josta näkee kaikkien polygonien rajat. Tätä kuvatiedostoa käytetään hyödyksi itse teksturointiin. Kun tekstuureja lähdetään tekemään, täytyy niissäkin pitää mielessä alusta, jolle peli on tulossa sekä mallin tärkeys. Tekstuuri kartan leveys ja korkeus on yleensä kahden potensseja, lähtien 32 px aina 8192 px asti. Medieval Battlefieldissä käytetään hahmolle 256 px kokoista textuuria, ja yhtä 128 px kokoista tekstuuria hahmon aseille. Nämä mitat ovat täysin riittävät kun otetaan huomioon kuinka kaukaa

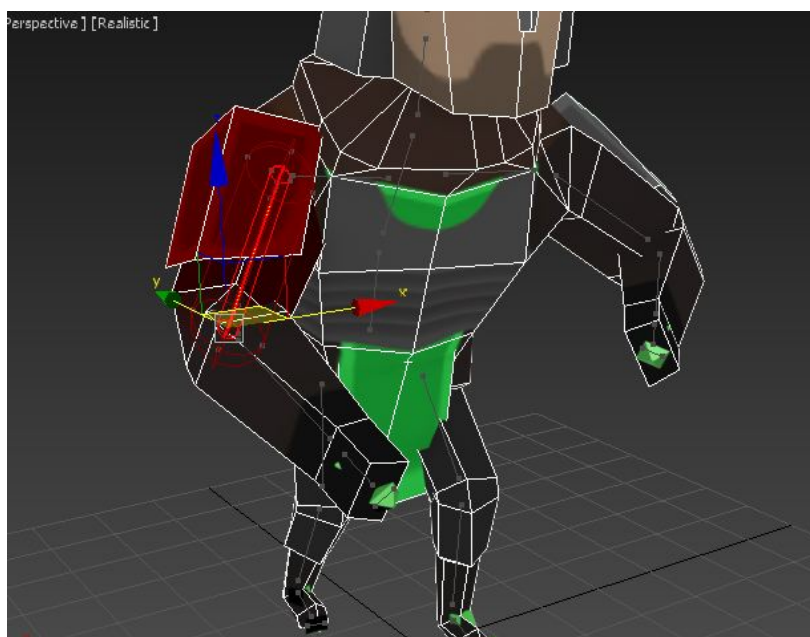
hahmojen malleja tarkkaillaan. Kuviossa 15 näkyy footman -hahmon UVW-mappaus 3ds Maxilla toteutettuna.



Kuvio 15. UVW mappaus 3ds maxissa

Kun malli on saatu teksturoitua, on sille tehtävä animaatiot, mikäli malli niitä tarvitsee. Animoiminen aloitetaan piirtämällä mallille luut, jotka sitten skinnataan malliin kiinni. Tämä onkin monen 3D-mallintajan mielestä se ikävin vaihe, sillä skinnaaminen harvoin onnistuu ensi yrittämällä. Skinnaaminen tapahtuu siten, että jokaista luuta varten kehittäjä asettaa vaikutusarvon jokaiseen vaikutettavaan polygonin pisteeseen. Skinnaaminen on helpompaa jos jo mallinnusvaiheessa on huomannut ottaa huomioon mihin mallin nivelet tulevat. Nivel kohtiin on hyvä käyttää muutama ylimääräinen polygoni, jotta nivelkohdat eivät niin sanotusti painu kasaan, kun niistä käännetään. Kuviossa 16 on Medieval Battlefieldin footman -hahmo skinnattu. Kuviossa on valittuna hahmon oikean käden olkaluu, jonka takia näkyy punaisella värillä polygonit, joihin tämän luun liikuttelu vaikuttaa. Skinnattu malli on valmis animoitavaksi. Animointi on helpointa tehdä 3ds Maxissa käyttämällä Autokey-ominaisuutta. Tällöin käyttäjä voi suoraan vaihdella eri ruutujen välillä, ja kääntää haluamansa osan haluttuun asentoon, ja 3ds Max laskee kaikki asennot välille jääviin ruutuihin. Tässä esitelty tapa hahmon animoimiseen on ehkä se helpoin

tapa animoida, mutta ei varmasti järkevin eteenkin, jos mallille on tarkoitus tehdä useita animaatiota. Monipuolisempaa animointia varten on hahmon nivelten liikkeet rajattava tiettyjen asteiden välille, sekä luille on annettava ”helppereitä”, jotka on linkitettyjä tiettyihin luihin. Näitä helppereitä liikuttelemalla saadaan nopeasti haluttua liikettä esimerkiksi raajoihin, kun käyttäjän ei tarvitse animoida muuta kuin helpperin liike ja 3ds Max laskee kaikki kyseiseen helpperiin liitettyjen luiden liikkeet. Helppereiden käyttö siis nopeuttaa itse mallin animointia ja animaation muokaamista, mutta on hieman työläämpi alustaa.



Kuvio 16. Skinnattu footman

Kun malli on mallinnettu, teksturoitu ja animoitu, voidaan siitä viedä kehitysovellukselle sopiva tiedosto. Tähän tiedostoon pakkautuu kaikki mallin animaatiot, skinit ja luut. Tekstuuri tulee usein erillisenä tiedostona vientikansioon.

5 Ansaitseminen pelillä

5.1 Pelien eri ansaintamenetelmät

5.1.1 Kertamaksuiset pelit

Kertamaksuiset pelit on pisimpään markkinoilla ollut ansaintamuoto pelinkehittäjille. Kertamaksuisuus on nykyäänkin suosittu tapa ansaita peleillä. Kertamaksuiset AAA-pelit maksavat julkaistessaan 40-70 euroa, riippuen alustasta ja julkaisijasta. Indiepelien hinta taas vaihtelee jopa kahdesta eurosta 30 euroon. Usein kertamaksuisiin peleihin julkaistaan pelin virallisen julkaisun jälkeen lisäosia, mikäli alkuperäisen pelin suosio on hyvä. Näiden lisäosien hinta vaihtelee myös viidestä eurosta 30 euroon, taas riippuen julkaisijasta ja lisäosan sisällöstä.

5.1.2 Kausimaksuiset pelit

Kausimaksuiset pelit ovat nykyään harvemmassa, mutta niitäkin yhä julkaistaan. Suurin osa kausimaksuisista peleistä on MMORPG -pelejä, eli valtavia nettiroolipelejä. Näissä peleissä on usein myös täysin ymmärrettävää, että pelaajalta peritään esimerkiksi 10 euroa kuukaudessa, koska pelin palvelimien ylläpitäminen ei ole halpaa kehittäjälle.

Kuuluisin kausimaksuinen peli on World of Warcraft. Se on julkaistu jo vuonna 2004, ja saavutti jo silloin suuren suosion. Pelissä on ollut koko sen elinkaaren ajan noin 12 euron kuukausimaksu, jota on voinut maksaa luottokortilla tai kaupoista saatavilla korteilla. Suosituimpina aikoina World of Warcraftilla oli jopa 12 miljoonaa maksavaa pelaajaa, joka teki suoraan 120 miljoonan dollarin kuukausituotot. On siis sanomattakin selvää ettei pelin kehittäjä, Blizzard, ole halunnut luopua kuukausimaksusta. World of Warcraftiin on myös julkaistu jatkuvasti lisäosia, jotka ovat käytännössä katsoen olleet pakollisia ostaa mikäli haluaa edetä pelissä muiden pelaajien kanssa. Nämä noin 2 vuoden välein julkaistut maksulliset lisäosat maksavat

noin 30 euroa julkaistessaan. Kun suurin osa aktiivisista pelaajista ostaa julkaistun lisäosan, tuo se mittavan lisätienestin pelin kehittäjille.

Eve Online on myös kausimaksuinen avaruussimulaatiopeli. Kuukausi peliaikaa maksaa 14,95 € (halvempi mikäli ostaa kerralla enemmän). Pelaaja pystyy kuitenkin halutessaan tienaamaan pelin sisällä kausimaksuun menevät rahat. Jos pelaaja on erittäin ahkera, voi hän jopa tienata hieman ylimääräistäkin pelaamalla. Mitään päivätyön korvaajaa siitä ei kuitenkaan voi saada, joten kyse ei ole mistään suurista summista. Eve Onlinen ekonomia on kuitenkin mielenkiintoinen vaihtoehto kausimaksuiselle pelille.

Nykyisin kuitenkin kausimaksuiset pelit ovat vähenemään päin. Moni peli, kuten Rift ja Tera, ovatkin vaihtaneet ansaintamalliaan julkaisun jälkeen. Molemmat pelin olivat julkaisussaan kausimaksuisia, 10 euroa kuukaudessa. Kuitenkin huomattuaan ettei pelit ole saanut kehittäjien haluamaa suosiota, on kausimaksuisuus korvattu sisältömaksuisuudella. Nykyisin pelit on täysin ilmaiseksi kokeiltavissa ja pelattavissa, mutta mikäli pelaaja haluaa esimerkiksi kosmeettisia päivityksiä hahmolleen on hänen käytettävä oikeaa rahaa. Tästä voidaan melko suoraan vetää johtopäätös, että kausimaksuiset pelit alkavat olla jo saapumassa suosionsa päähän.

5.1.3 Sisältömaksuiset pelit

Sisältömaksuiset pelit ovat kehittäjille nykyaikana todennäköisesti yleisen tapa ansaita rahaa peleistään. 1990- ja 2000-luvun alussa voitiin vielä ajatella, että sisältömaksuiset pelit olivat pääsääntöisesti pelejä, joihin julkaistiin pelin julkaisun jälkeen maksullisia lisäosia, jotka toivat käyttäjälle jotain uutta sisältöä pitämäänsä peliin. Tästä syystä vielä ennen 2010-lukua sisältömaksuiset pelit oli samassa kategoriassa kertamaksuisten pelien kanssa.

Nykyiset sisältömaksuiset pelit eivät usein maksa mitään pelaajalle aloittaakseen pelaamista, vaan pelin ansainta saadaan kun pelaajat ovat jo jääneet peliin ”koukkuun” ja saada pelistä enemmän irti sekä samalla mahdollisesti haluavat tukea

pelin kehittäjää. Nykyään sisältömakuiset pelit ovatkin erityisesti suosittuja mobiilipelien kehittäjien keskuudessa. Tämä johtuu siitä, että mobiilipelikaupat ovat yhä täydempiä uusista peleistä, eikä kuluttaja välttämättä halua sijoittaa rahaansa peliin, josta ei voi olla varma, että onko se hyvä. Näin kehittäjä tarjoaa riskittömän vaihtoehdon pelaajalle kokeilla peliä. Sisältömakuisissa mobiilipeleissä ansainta saadaan pääsääntöisesti käyttäjän pelikokemusta helpottavilla tai pidentävillä komponenteilla. Nämä komponentit usein on esimerkiksi pelin sisäistä valuuttaa, jota voidaan käyttää vaikkapa oman pelaajan lisäämiseen tai ostaakseen käyttäjälle tavaroita, jotka auttavat pelaajaa päihittämään pelaajat, jotka eivät ole sijoittaneet rahaansa pelin sisältöön. Hyvä esimerkki sisältömakuisen mobiilipelin menestymisestä on Clash of Clans -strategiapeli. Siinä ansainta saadaan pelin sisäisestä valuutasta, jalokivistä. Näitä jalokiviä voi ostaa eri suuruksissa paketeissa, joista suurimmissa paketeissa saa paljousalennusta.

Myös tietokonepeleissä on yleistynyt sisältömakuiset pelit. Poiketen kuitenkin mobiilialustoille suunnatuista sisältömakuisista peleistä, tietokoneelle julkaistut sisältömakuiset pelit ovat usein lähinnä pelaajan pelin etenemistä nopeuttavia apuja. Pelaajien keskuudessa usein suositaankin sellaisia sisältömakuisia pelejä, jotka eivät ole niin sanotusti P2W eli Pay to win -pelejä. Esimerkiksi tietokoneelle jo useita vuosia sitten julkaista Counter Strike: Global Offensive -peli saa suuren osan tuotostaan peliin ostettavien kosmeettisten ase-skinien muodossa. Näitä skinejä saa pelistä putoavista laatikoista, joiden aukaisemiseen tarvitaan avain, joka maksaa noin 2,3 euroa. Pelaaja voi kuitenkin myydä saamansa ase-skinin pelin marketissa oikeaa rahaa vastaan. Pelin kehittäjä kuitenkin ottaa noin 20% osuuden koko kauppahinnasta ja ansaitsee myös sitä kautta pelillään. Peliin onkin muodostunut oma ekonomiansa, jossa harvinaiset ase-skinit voivat maksaa jopa tuhansia euroja.

Sisältömakuiset pelit onkin nykyään mahdollisesti suosituin ansaintamuoto kehittäjien ja julkaisijoiden keskuudessa.

5.2 Mikä on käyttäjäystävällisin tapa ansaita pelillä

Käyttäjäystävällisestä tavasta ansaita peleillä voidaan olla montaa eri mieltä. Tämän hetkisiä keskusteluja seuratessa voi huomata, että pelaajat itse suosivat, tai ainakin toivovat kehittäjiltä, sisältömaksuisia pelejä. Koska pelejä julkaistaan nykyään niin monelle eri alustalle ja niin monessa eri kaupassa, on käyttäjälle turvallisin malli kokeilla peliä ilmaiseksi ja maksaa sitten pelin sisällöstä halutessaan. Usein jos peli onkin hyvin toteutettu ovat pelaajat valmiita käyttämään myös hieman rahaa saadakseen pelistä enemmän irti. Useammin pelaajat ovat suotuisampia maksullista sisältöä kohtaan jos se ei tarjoa ylivoimaa maksaville pelaajille, vaan maksullisen sisällön tulisi olla enemmän kosmeettista tai pelin kulkua nopeuttavaa.

Keskustelupalstoja seuraamalla voi havaita, että käyttäjät eivät ole kovin innoissaan Pay to win -peleistä. Vaikuttaisi kuitenkin, että nuoremman sukupolven pelaajat ovat valmiita myös maksamaan pärjätäkseen pelissä paremmin. Tämä oletettavasti johtuu siitä, että nuoremmat pelaajat ovat kasvaneet aikakaudella, jolloin peleihin alettiin sisällyttämään P2W -ominaisuuksia, ja tätä kautta ovat hyväksyväisempiä tälle menettelylle.

Jos kuitenkin etsitään sitä käyttäjäystävällisintä tapaa ansaita pelillä, voidaan melko yleistäen sanoa, että käyttäjät hyväksyvät parhaiten joko kertamaksuisen pelin, jolloin he saavat koko pelin pelattavakseen yhdellä kiinteällä hinnalla. Toisena suosittuna ja käyttäjäystävällisenä menetelmänä voidaan pitää ilmaisia, tai kohtuullisen kertamaksun sisältäviä sisältömaksuisia pelejä, joissa ei kuitenkaan tarjota rahaa vastaan pelaajalle ylivoimaa muiden pelaajien päihittämisessä.

5.3 Millä tavalla ansaitsee parhaiten?

Tähän kysymykseen on haastavaa saada suoraa vastausta, ansaintamalli ei kuitenkaan ole ainoa asia joka vaikuttaa kuinka paljon pelillä ansaitsee. Koska tarkkaa tietoa pelien tuotoista ei löydy netistä, ihan ymmärrettävästä syystäkin, on vertailun luvut keräilty useista lähteistä eikä tästä syystä ole välttämättä kovin tarkkoja.

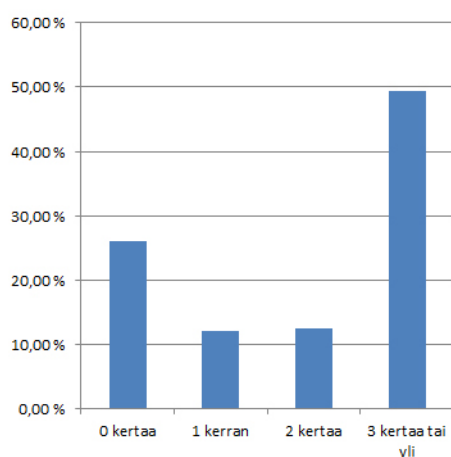
Vertailua voidaan kuitenkin pitää suuntaa-antavana. Vertailuissa on oletuksena, ehkä hieman liioitellusti, että keskiverto pelaaja pelaa peliä vuoden verran ennen sen unohtamista. Jokainen arvio on laskettu tuhannella pelaajalla.

Guild Wars 2 on kertamaksuinen nettiroolipeli, jossa on myös mahdollisuus sisältömaksullisuuteen. Kun vertailu tehdään tuhannella pelaajalla, voidaan suoraan laskea, että tuhat pelaajaa on ostanut pelin. Pelistä on saatavilla myös deluxe-versio, joka on hieman kalliimpi. Tarkkaa tietoa siitä, kuinka moni on ostanut deluxe-version. Internetistä löytyi eri peleistä käyttäjien kyselyitä ketkä ostanut, mutta koska kysymys oli usein myös esitetty muodossa ”Ketkä ostivat deluxe-version?”, oli myös vastaajat yleensä vain niitä, jotka oli ostanut deluxe-version. Tästä syystä kyselyiden vastauksia on kompensoitu ottamalla huomioon kuinka moni on lukenut kyselyn, sillä oletettavasti he ketkä eivät vastanneet, eivät myöskään ostaneet deluxe-versiota. Nämä seikat huomioon ottaen saatiin deluxen ostaneiden ja ei ostaneiden suhteeksi 1/9. Tästä voidaan laskea, että tuhannesta pelaajasta 112 oli ostanut deluxe-version. Pelin hinnoissa ei ole suuria muutoksia ollut pelin elinkaaren aikana, joten keskiarvohinta on laskettu toukokuun 2015 hintahaarukkaan. Normaalin version keskiarvohinnaksi saatiin 34,3 euroa ja deluxe-version keskiarvohinnaksi 51,7 euroa. Näillä tiedoilla voidaan arvioida kehittäjien tienanneen pelkästään pelien aloitusmaksuista noin 36 250 euroa, joka tekee 36 euroa ja 25 senttiä per henkilö.

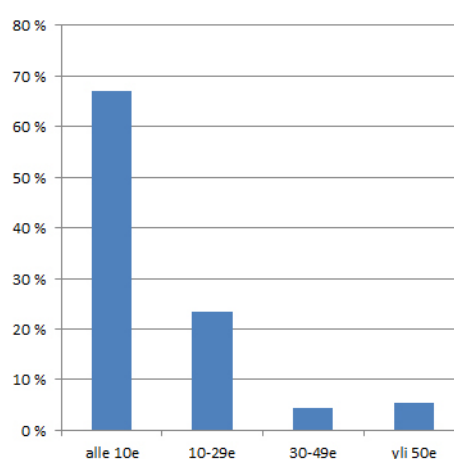
Koska Guild Wars 2 kaikki tulot eivät kuitenkaan rajoitu ainoastaan pelaajien maksamiin kertamaksuihin, on laskettava mukaan myös pelin sisäisiin ostoksiin käytetty raha. Kuviossa 17 on hasweL nimisen Reddit-sivuston käyttäjän tekemän kyselyiden vastaukset, kyselyihin vastasi yhteensä 655 henkilöä. Kyselyiden mukaan valtaosa pelaajista käyttää alle 10 euroa kuukaudessa Guild Wars 2:n maksulliseen sisältöön. Noin 24 prosenttia käyttää 10 – 29 euroa kuussa, 30 – 49 euroa käyttää noin 5 prosenttia sekä noin 5 prosenttia käyttää yli 50e. Pelaajilta kysyttäessä, kuinka usein he ovat ostaneet pelin sisäistä materiaalia, noin puolet pelaajista oli ostanut yli 3 kertaa. 2 kertaa ostaneita joukossa oli vajaa 13 %, yhden kerran ostaneita reilu 12 % ja 0 kertaa ostaneita 26 %. Tästä voimme jo laskea että 26 prosenttia pelaajista ei

ole käyttänyt enempää rahaa peliin kuin alkusijoituksen. Koska on mahdoton näiden kahden eri kysymyksen väliltä verrata loppuja ostoksia, on yksi ja kaksi kertaa ostaneiden pelaajien ostosummat suhteutettu suoraan kysymykseen ”Paljonko käytät kuukaudessa?”. Guild Warsin minimi ostos on 10 euroa, joten yhden ja kaksi kertaa ostaneiden kohdalla voidaan olettaa alle 10 euroa vastauksen tarkoittavan tasan kymmenen euroa. Näin saadaan selville, että tuhannen henkilön joukosta 240 henkilöä käyttää pelin aloitusmaksun lisäksi peliin noin 3790 euroa. Lopusta 493 henkilöstä tuhannesta on ostanut kolme kertaa, tai useammin. Ostokerrat on laskettu niin, että oletetaan keskiarvo pelaajan maksaneen noin kuutena kuukautena vuodessa. Tällä oletuksella laskettuna useammin kuin kaksi kertaa peliin sijoittaneet kuluttavat vuodessa noin 6 355 euroa. (haswel 2014)

Kuinka usein olet ostanut pelin sisäistä valuuttaa?



Kuinka paljon käytät peliin kuukaudessa?



Kuvio 17. Guild Wars 2 sisältöostokyselyn tulokset (haswel 2014)

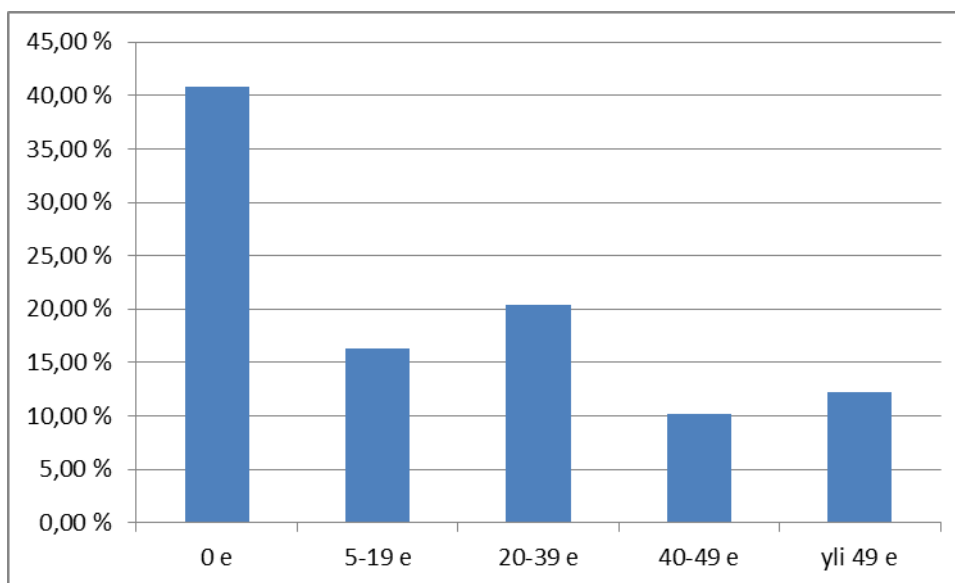
Kun kaikki maksulliseen sisältöön käytetty lasketaan yhteen, voidaan olettaa tuhannen pelaajan käyttävän noin 10 145 euroa vuodessa peliin. Tämä antaa keskiarvoksi henkilöä kohden 10 euroa ja 15 senttiä.

World of Warcraft on huomattavasti helpompi peli ottaa vertailuun. Sen aloitusmaksu vaihtelee 59,99 ja 15 euron välillä, riippuen haluaako pelaaja ostaa pelin perusversion, joka ei pidä sisällään viimeisintä lisäosaa, lisäosallisen

perusversion vai deluxe-version. Voidaan taas olettaa, että deluxe-versiota pelistä ostaa noin yksi yhdeksästä. Tällöin 112 pelaajaa käyttää pelin aloittamiseen 6 160 euroa. Oletettavasti ns. karvalakkiversion osuus myynnistä on hyvin pieni, oletetaan, että noin 50 henkilöä tuhannesta on ostanut sen. Tämä tekee yhteensä 750 euroa. Loput pelaajat ovat siis näillä laskuilla ostaneet pelin perusversion, jonka keskihinta on 37,5 euroa. Loput pelaajat siis sijoittavat yhteensä 31 425 euroa, ja kun kaikki hintaluokat lasketaan yhteen, tekee se 38 335 euroa aloitusmaksuista tuhannelta henkilöltä. Henkilöä kohden aloitusmaksu on keskimäärin 38,33 euroa.

Koska World of Warcraftin kuukausimaksu vaihtelee riippuen siitä kuinka paljon peliaikaa kerralla ostaa, on kuukausikulu laskettu kuukausimaksujen keskiarvolla, eli 11,99 euroa. Tästä voidaan suoraan laskea, että vuodessa pelaaja maksaa noin 143,88 euroa pelataksaan peliä. Tuhannella henkilöllä se tekee 143 880 euroa vuodessa. World of Warcraftissa on myös maksullisia ominaisuuksia, kuten hahmon siirto serveriltä toiselle tai hahmon nimen vaihtaminen. Nämä palvelut maksavat noin 10 euroa kappale, mutta niihin käytettyjä rahoja ei ole laskelmissa otettu huomioon.

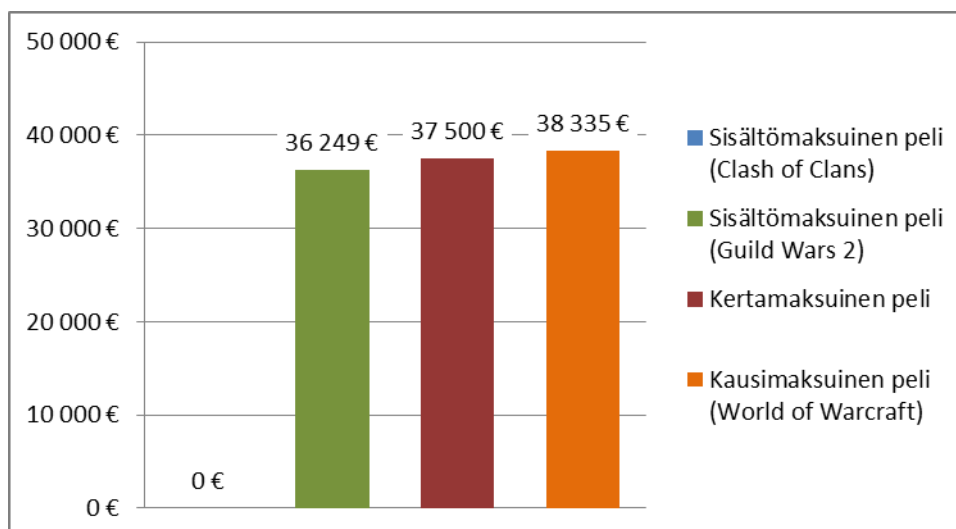
Clash of Clans on ilmainen mobiilipeli, jossa on maksullinen sisältö. Clash of Clans voitaisiin jopa luokitella P2W-peliksi, sen hitaan etenemisen takia, joka pelaajan menestystä hidastaa mikäli hän ei ole valmis maksamaan boostereista tai muista pelin sisäisistä tavaroista. Koska peli on ilmaiseksi ladattavissa mobiilikaupoista, ei sillä ole lainkaan aloitusmaksua. Koska tarkkaa tilastoa ei löytynyt siitä kuinka paljon keskivertopelaaja käyttää Clash of Clans -peliin, haettiin tähän tarvittava tieto netin keskustelupalstoilta. Vertailuun saatiin yhteensä 49 vastausta, joiden keskiarvoksi saatiin noin 19 euroa. Kuviossa 18 on tarkemmin jaettu vastanneiden käyttämät summat. 19 euron keskiarvo on varsin odotettava summa, sillä pelin ainoa pitkällä tähtäimellä järkevä sijoitus, rakentajat, maksavat noin 20 euroa aukaista. Näillä pelaaja saa nopeutettua pelinsä etenemistä huomattavasti. Vuositasolla tuhat pelaajaa käyttää noin 19 387 euroa Clash of Clans -peliin.



Kuvio 18. Clash of Clans pelaajien käyttämät rahasummat

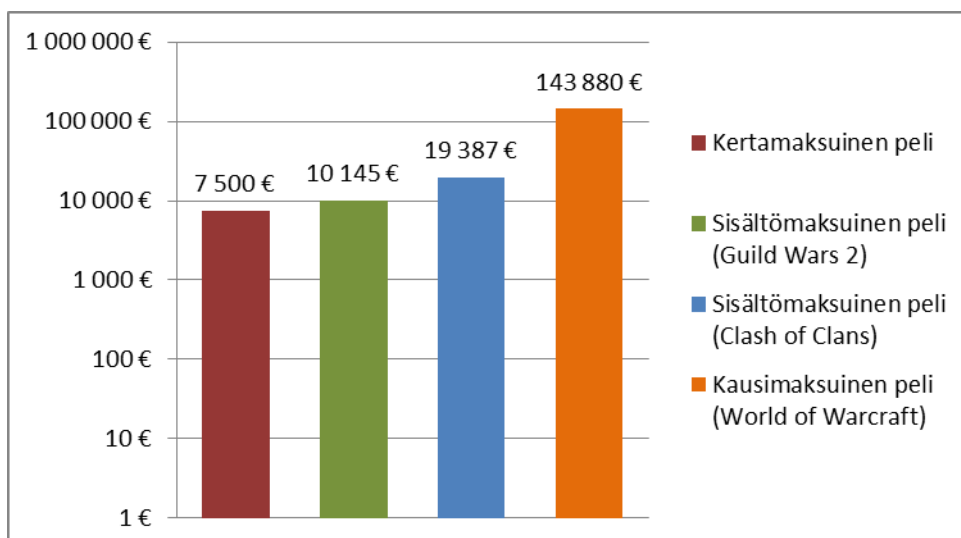
Kertamaksuisen pelin keskiarvohinnaksi voidaan laskea noin 37,5 e. Tällöin aloituskustannus pelille on tuhannelta henkilöltä 37 500 euroa. Jos oletetaan, että tavanomainen kertamaksuinen peli julkaisee vuoden sisällä kaksi lisäosaa pelilleen, ja julkaistun lisäosan keskiarvoinen hinta on noin 10 euroa. Oletetaan, että puolet pelaajista ostaa ainakin yhden lisäosan ja neljäsosa molemmat lisäosat, tulee vuodessa tuloja 7 500 euroa tuhannelta henkilöltä. Tämä tarkoittaa, että keskimääräinen pelaaja käyttää kertamaksuisen pelin lisäosiin noin 7,5 euroa vuodessa.

Kun eri ansaintamuotojen aloitusmaksuja lähdetään vertailemaan keskenään, nähdään, että kehittäjän tuloissa ei ole merkittävää eroa, mikäli he päättävät peliltään periä aloitusmaksua, lukuun ottamatta Clash of Clansia, joka ei ilmaisella pelillä luonnollisestikaan saa tuloja aloitusmaksuista. Kuviossa 19 on asetettu vertailuun eri tyyppisten pelien aloitusmaksut, Clash of Clansin kohdalla tulos on nolla euroa, koska pelillä ei ole latausmaksua.



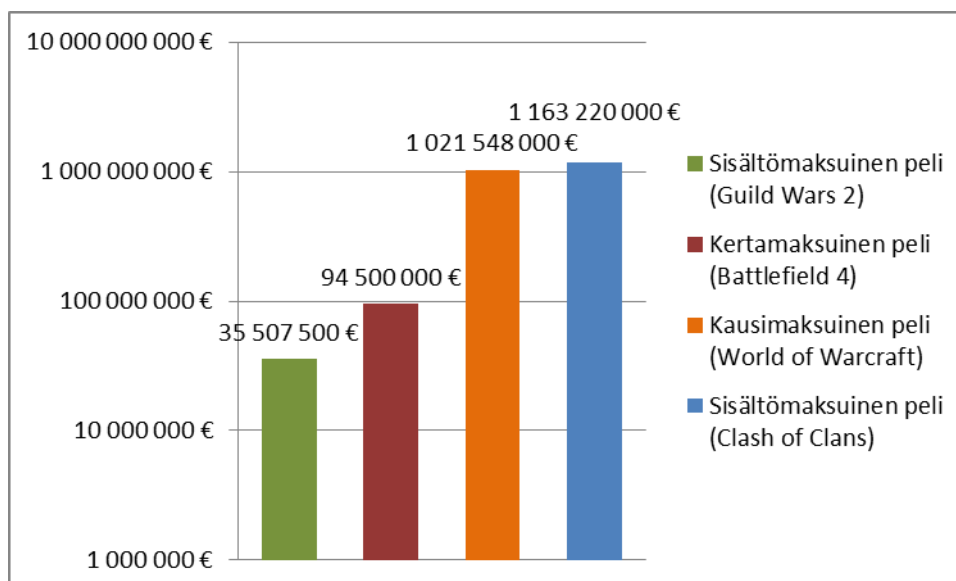
Kuvio 19. Eri ansaintamuotojen pelien tuomat tulot aloitusmaksuista tuhatta henkilöä kohden

Kun taas vertaillaan eri ansaintamuotojen tuomia tasaisia tuloja vuoden ajalta, alkaa ansaintamuodoissa olla selkeää eroa. Kuviossa 20 on vertailtu eri ansaintamuotojen tuomat tulot vuoden ajalta. Tästä vertailusta voimme selkeästi todeta, että kausimaksuinen peli tuottaa parhaiten vuoden aikana tuhatta pelaajaa kohden. Toisena ja kolmantena tulee sisältömaksuiset pelit. Koska tämä vertailu on tehty pitkälti oletusten pohjalta, ei se näytä koko totuutta, mutta se toimii kuitenkin varmasti suuntaa antavana vertailuna. Myöskin vertailussa on otettava huomioon, että vertailuun otetut pelit ovat jokainen erilaisia, ja tästä syystä ei täysin suoraan verrattavissa toisiinsa. Lisäksi laskelmissa ei ole ollenkaan otettu huomioon pelien ylläpitokustannuksia, kehittäjien palkkoja, mobiilikauppojen ottamia osuuksia, veroja, valmistusmaksuja jne.



Kuvio 20. Eri ansaintamuotojen tuomat tulot vuoden ajalta tuhatta henkilöä kohden (huomioi logaritminen asteikko)

Kun vuoden tulot kerrotaan pelien pelaajamäärillä, sekä kertamaksuinen peli myyntiluvuilla, saadaan jo hieman parempaa osviittaa siitä minkä verran kukin peli tuo tuloja vuoden aikana. Kuviossa 21 on tehty tämä vertailu, Guild Wars 2 sekä Clash of Clans eivät ole kumpikaan julkaisseet tarkkaa tietoa pelaajamäärästä, joten näissäkin on jouduttu nojaamaan netistä löytyviin arvioihin. Clash of Clans Google play kaupan lataukset näyttivät kirjoitushetkellä 9.5.2015 100-500 miljoonaa latausta, pelin vuoden aikaiseksi aktiiviseksi pelaajakunnaksi kaikilla alustoilla arvioitiin yhteensä 60 miljoonaa pelaajaa. Guild Wars 2:ta ei löytynyt myöskään tarkkaa tietoa, mutta julkaisustaan neljässä kuukaudessa peliä myytiin 3,5 miljoonaa kappaletta. Tätä lukua on suoraan käytetty arviona aktiivisista pelaajista vuoden aikana. Kertamaksuisen pelin kohdalla on vertailua varten tarkistettu Battlefield 4 pelin myyntiluvut, 12,6 miljoonaa kappaletta.



Kuvio 21. Pelien vuositulot koko pelaajamäärällä

Kun kuvion 18 tuloksia vertaillaan internetistä löytyviin julkisiin vuositulotietoihin, osaa laskelmat hyvin lähelle yritysten virallisesti antamia tuloksia.

5.4 Oman pelin ansaintaperiaate

5.4.1 Freemium-menetelmä

Pelin (Medieval Battlefield) ansaintamenetelmä on nykytrendin mukaan freemium. Freemium on liikemalli, jossa käyttäjälle tarjotaan itse tuote ilmaiseksi, mutta jos käyttäjä haluaa tuotteestaan enemmän irti, on hänelle tarjolla korvausta vastaan uusia ominaisuuksia tuotteelleen. Tämä tarkoittaa Medieval Battlefieldin kohdalla sitä, että pelin kaikki sisältö on ilmaiseksi saatavilla, ja pelattavissa. Mutta mikäli pelaaja haluaa edetä pelissä nopeammin ja saada uusia esineitä, joukkoja yms. auki nopeammin, hänen on käytettävä oikeaa rahaa.

Pelaaja voi käyttää rahaansa ostaakseen pelin premium-valuutta, jota voi käyttää pelin sisällä esimerkiksi boostereiden ostoon, joilla voidaan nopeuttaa kertyviä kokemuspisteitä tai rahaa. Ansaintamenetelmiä verratessa huomattiin, että Clash of Clansin tapa tarjota pelaajalle noin 20 eurolla pysyvä hyöty pelille on monien

pelaajienkin hyväksymä. Clash of Clansissa voidaan kohtuullisella rahasummalla ostamaan rakentajia, jotka nopeuttavat pelaajan etenemistä pelissä. Näitä rakentajia pelaajalta ei viedä missään pelin vaiheessa, vaan ne ovat pysyvät. Tästä syystä myös Medieval Battlefieldiin päätettiin lisätä parantajat, jotka ovat pelaajan kasarmilla nopeuttamassa joukkojen taistelukunnon palautumista. Näin pelaaja voi sijoittaa peliin pienen summan rahaa, ja saada vastineeksi boostereita tai palautusjuomia pysyvemmän sijoituksen. Pelaajalle tarjotaan pelin alussa yksi parantaja, loput 4 parantajaa voidaan ostaa suoraan pelin premium-valuutalla, tai pelaaja voi vaihtoehtoisesti useiden viikkojen pelaamisella avata tarvittavan määrän premium-valuutta loppujen parantajien avaamiseksi.

Toinen harkittu ominaisuus on myös sepän lisääminen peliin. Sepät ovat hahmoja, joita voidaan käyttää luomaan erityisen tehokkaita aseita omien joukkojen käyttöön. Seppiä ei aluksi ole pelaajalle tarjolla, mutta pelin edetessä hänellä on mahdollisuus niitä ostaa. Seppiä voi olla maksimissaan kaksi kappaletta, joista vain toisen voi aukaista ilman rahan sijoittamista peliin, tällöinkin muutamien viikkojen pelaamisella.

5.4.2 Mainokset

Toinen tapa pelillä ansaitsemiseen on mainokset. Usein mobiilipeleissä nähdäänkin pieniä mainoksia pelin alalaidassa ja joskus isompia koko ruudun kokoisia ohitettavia mainoksia. Pelissä on ainakin pieniä mainosbannereita, mutta koko ruudun kokoiset ohitettavat mainokset on harkinnassa.

6 Pohdinta

Opinnäytetyön tavoitteena oli tuoda tietotaitopohja pelien kehittämiseen projektin toteuttajalle, sekä luoda toimiva pohja pelille, joka opinnäytetyötä varten aloitettiin. Vaikkakin itse peli jäikin lähinnä ”proof of concept” -tasolle, saatiin siitä hyvä pohja kesällä ja syksyllä tapahtuvaa kehitystä varten. Alkuperäisen suunnitelman mukaan tämän opinnäytetyön aikana oli tarkoitus kehittää Medieval Battlefield jo niin pitkälle, että sitä pystyisi pelaamaan kaksi pelaajaa omilla puhelimillaan internetin välityksellä. Valitettavasti aikarajoitteet tuli vastaan tässä, eikä kaikkia suunniteltuja ominaisuuksia ehditty toteuttamaan.

Vaikkakaan lopputulos ei laajuudeltaan ollut aivan sitä mitä alun alkaen projektilta odotettiin, oli kuitenkin jo tuotettu osuus erittäin opettavaista, ja haettua kokemusta saatiin monelta osa-alalta. Opinnäytetyön ulkopuoliseksi opeteltavaksi jäi lähinnä verkko-ominaisuuksien toteuttaminen sekä valikoiden luominen järkevästi. Opinnäytetyön tutkielmissa saatiin hyvää opetusta siitä, kuinka pelin toteuttamista kannattaa järkevästi lähestyä.

Pelintekoa tutkiessa prosessina, saatiin paljon arvokasta oppia siitä mitä peliä suunnitellessa ja toteuttaessa on syytä ottaa huomioon. Etenkin suunnitteluvaiheessa olevia asioita listatessa tuli opinnäytetyö projektin kehittäjälle selkeämmäksi mitä olisi pitänyt ottaa huomioon ja mitä asioita teki oikein. Toteutusvaiheen tutkimisesta selvisi myös muutamia tarkasteltavia ongelmakohtia, näitä oppeja ei kuitenkaan päästy opinnäytetyön projektin puitteissa paljoa hyödyntämään. Läpikäytyjä testausmenetelmiä tullaan varmasti hyödyntämään kun pelin kehitystä jatketaan. Tällä toivottavasti saavutetaan pelille parempi toimivuus julkaisuvaiheessa.

Itse pelimoottorien vertailu ei sinänsä tarjonnut liiemmin uutta tietoa pelin kehitystä ajatellen, mutta se kuitenkin osoitti hyvin sen, että pelimoottori valinta oli hyvä tälle projektille.

Todellinen hyöty opinnäytetyöstä oli pelien ansaintamenetelmien vertailu. Se osoitti hyvin sen mikä ansaintamenetelmä käy missäkin tilanteessa, mitä käyttäjät haluavat ja mikä on tuottoisin. Vertailu eri ansaintamenetelmien välillä olikin erittäin mielenkiintoista itse opinnäytetyön kirjoittajalle ja se tarjosi hedelmällistä tietoa siitä miten paljon menestyvät pelit oikeasti tuottavat. Tätä tutkimusta oltaisiin voitu tuki jatkaa pidemmälle tutkimalla kuinka suuret yritysten sijoitukset pelien kehittämiseen olivat ja tällä tavoin selvitetty kuinka suuret voitot tuotteet antavat. Näillä tiedoilla olikin helppo valita opinnäytetyön projektille sopivaksi ansaintamenetelmäksi sisältömaksuisuus.

Tällä opinnäytetyöllä saavutetulla kokemuksella on erittäin hyvä lähteä kehittämään aloitettua projektia eteenpäin. Opinnäytetyön tarjoamalla opeilla toivottavasti onnistutaan saamaan pelille edes jonkinlainen menestys ja sitä kautta tuloja kehittäjälle.

Lähteet

Clickteam. 2015. Wikipedia-sivusto. Viitattu 14.4.2015.

<http://en.wikipedia.org/wiki/Clickteam>

Dereau, B. 2015. Unreal Paris – virtual tour. Dereau Benoit –CV. Viitattu 25.4.2015.

<http://www.benoitdereau.com/unrealparis.html>

Derk. 2012. The 8-Step Guide To Interface Design for iPhone Games. Paladin Studios-sivusto. Viitattu 3.5.2015.

<http://www.paladinstudios.com/2012/04/23/the-8-step-guide-to-interface-design-for-iphone-games/>

GameMaker: Studio. 2015. Wikipedia-sivusto. Viitattu 15.4.2015.

http://en.wikipedia.org/wiki/GameMaker:_Studio

hasweL. 2014. How much do you all spend on gems? Reddit –sivusto. Viitattu 8.5.2015.

http://www.reddit.com/r/Guildwars2/comments/2bj6n6/how_much_do_you_all_spend_on_gems/cj5urcv

Kumar, V. 2014. Making “Freemium” Work. Harvard Business Review-sivusto. Viitattu 7.4.2015.

<https://hbr.org/2014/05/making-freemium-work>

p.a.m.e.l.a. 2015. p.a.m.e.l.a game. pamelagame.com –sivusto. Viitattu 25.5.2015

<http://www.pamelagame.com/>

Tpelgam42. 2014. UDK vs. Unreal Engine 4. Cave in –blogi. Viitattu 25.4.2015.

<http://caveingame.com/udk-vs-unreal-engine-4/>

Unity (game engine). 2015. Wikipedia-sivusto. Viitattu 13.4.2015.

http://en.wikipedia.org/wiki/Unity_%28game_engine%29

Unreal Engine. Wikipedia-sivusto. 2015. Viitattu 15.4.2015.

http://en.wikipedia.org/wiki/Unreal_Engine

Ville-Veikko, H. 2014. Mobile Game Testing – Part #1: The Importance and Difference from App Testing. Testdroid –sivusto. Viitattu 5.5.2015.

<http://testdroid.com/tech/mobile-game-testing-the-importance-and-difference-from-app-testing>